Note Generali Applicativo

Sommario

[Architettura generale del sistema 2](#_Toc67058589)

[Database “ad oggetti” 2](#_Toc67058590)

[OPENgov in dettaglio 2](#_Toc67058591)

[Moduli applicativi 3](#_Toc67058592)

[Applicativi 3](#_Toc67058593)

[OPENgov 3](#_Toc67058594)

[AnagraficaWEB 3](#_Toc67058595)

[Stradario 3](#_Toc67058596)

[OPENgovTERRITORIO 3](#_Toc67058597)

[DichiarazioniICI 3](#_Toc67058598)

[OPENgovTIA 3](#_Toc67058599)

[OPENgovTOCO 3](#_Toc67058600)

[OPENutenze 3](#_Toc67058601)

[OPENgovPROVVEDIMENTI 3](#_Toc67058602)

[Modulo Stampa 3](#_Toc67058603)

[Note Generali 3](#_Toc67058604)

[Librerie 4](#_Toc67058605)

[Base Dati. 4](#_Toc67058606)

[Funzioni applicative 5](#_Toc67058607)

[Problematiche 5](#_Toc67058608)

[Banca dati 5](#_Toc67058609)

# Architettura generale del sistema

OPENgov è raffigurabile utilizzando la metafora del puzzle composto da tessere eterogenee: database, librerie, moduli applicativi specializzati, windows services, black box.

L’ambiente operativo è naturalmente appoggiato su sistemi dotati di interfaccia grafica e di capacità di gestione “object oriented” a basso livello. Il sistema operativo di riferimento è pertanto “Microsoft Windows” nella sua evoluzione a 32 bit, in modalità web.

Graficamente il “puzzle” si configura in questo modo:



Tutto il team coinvolto nella progettazione e nello sviluppo di OPENgov dovrà quindi avere sempre mentalmente in primo piano alcune linee fondamentali “di filosofia” così riassumibili:

1. Qualsiasi collegamento tra un’applicazione ed il database va realizzata con componenti invisibili all’utente (black box).
2. Gli utilizzatori tra loro (e specularmente anche le applicazioni tra loro) devono poter “parlare” attraverso un sistema integrato interno ad OPENgov.
3. Gli elementi del database definiti come “di confine” (ossia oggetto di transazioni col mondo esterno al sistema) devono essere considerati e gestiti a livello “HAL”, liberi quindi da qualsiasi vincolo di qualsiasi natura legato a risorse hardware.

## Database “ad oggetti”

Realizzare dei “livelli di astrazione dalle applicazioni” (AAL) significa necessariamente concepire il database come un insieme di oggetti, ciascuno dotato di proprietà e metodi particolari.

Il documento di definizione del database OPENgov è quindi da considerarsi tassativamente vincolante a livello di oggetti definiti come tali; questo significa che qualsiasi eventuale modifica di struttura che tocchi uno o più campi “oggetto” deve necessariamente comportare una revisione totale di tutti i componenti ad esso collegati.

Significa anche che qualsiasi operazione su uno o più campi “oggetto” deve essere effettuata da un unico componente specializzato (“CreaCodiceFiscale”, “VisualizzaDocProtocollo”, “EditaTestoDelibera”, “CambioIndirizzoResidente”, eccetera).

Quando possibile e se necessario per ragioni di complessità delle interazioni scatenate, questi componenti specializzati dovranno essere realizzati come oggetti esterni e pertanto trasportabili (DLL, EXE, COM, eccetera); in tutti gli altri casi verranno scritti come “moduli” riusabili e condivisi dalle singole applicazioni (PRG in VFoxPro, BAS in VB, C, eccetera).

## OPENgov in dettaglio

L’architettura complessiva di OPENgov può essere schematizzata così:



Abbiamo quindi le seguenti tipologie di componenti:

1. Moduli applicativi
2. Librerie
3. Base dati

### Moduli applicativi

OPENgov è sviluppato in ambiente .NET su base WebForms. È l’insieme delle singole applicazioni, ciascuna destinata ad un compito specifico. La maggior parte delle interazioni verso il database vengono fatte tramite viste o stored procedure. I moduli applicativi “base” di OPENgov sono quelli qui di seguito elencati:

* Gestione servizi demografici
* Anagrafe territoriale comunale
* Anagrafe tributaria comunale
* Gestione Tariffa smaltimento Rifiuti Solidi Urbani
* Gestione Imposta Comunale sugli Immobili
* Gestione COSAP
* Gestione Acquedotto
* Gestione Dati Esterni

Il linguaggio di sviluppo utilizzato sarà indistintamente VB.NET e C#.

#### Applicativi

##### OPENgov

È il modulo di gestione dei dati generali del sistema come la login al sistema, gli enti in lavorazione, i dati dei residenti e le interrogazioni generali. Sviluppato in VB.NET, al suo interno sono referenziati tutti i singoli moduli applicativi.

##### AnagraficaWEB

È il modulo di registrazione dei dati anagrafici dei soggetti passivi dei tributi. Sviluppato in VB.NET.

##### Stradario

È il modulo di interrogazione dei dati dello stradario. Sviluppato in VB.NET. Si interfaccia al database tramite il servizio Stradario.

##### OPENgovTERRITORIO

È il modulo di registrazione dei dati del Territorio come vani, comuni e strade. Sviluppato in VB.NET.

##### DichiarazioniICI

È il modulo di registrazione dei dati ICI/IMU/TASI. Sviluppato in C#. Utilizza il servizio MotoreICI per il calcolo del dovuto.

##### OPENgovTIA

È il modulo di registrazione dei dati TARSU/TARES/TARI. Sviluppato in VB.NET. Utilizza il servizio MotoreTARSU per il calcolo del dovuto.

##### OPENgovTOCO

È il modulo di registrazione dei dati OSAP/SCUOLE/ICP. Sviluppato in C#. Utilizza il servizio MotoreOSAP per il calcolo del dovuto.

##### OPENutenze

È il modulo di registrazione dei dati dell’Acquedotto. Sviluppato in VB.NET. Utilizza il servizio MotoreH2O per il calcolo del dovuto.

##### OPENgovPROVVEDIMENTI

È il modulo di registrazione dei dati degli avvisi di Accertamento per tutti i tributi. Sviluppato in VB.NET. Utilizza il servizio MotoreProvvedimenti per alcune interrogazioni e il salvataggio degli avvisi creati.

##### Modulo Stampa

Non è un modulo WEB vero e proprio ma si compone di due servizi:

1. [ServizioElaborazioneDatiStampe](#_Servizio_ElaborazioneDatiStampe), il cui compito è prelevare i dati dai database e comporre l’elenco dei segnalibri da popolare
2. [ServizioStampe](#_Servizio_Stampe), il cui compito è stampare in Word o PDF l’elenco dei segnalibri ricevuti in ingresso.

#### Note Generali

Tutte le griglie utilizzate nel sistema sono un oggetto proprietario (Ribes.OPENgov.WebControls) che estende le proprietà ed i metodi della GridView standard.

La dll Utility contiene le funzioni di utilizzo generali, come ad esempio la conversione nei vari tipi di dato gestendo l’opzione null; si occupa inoltre di gestire le connessioni verso il database.

Il sistema utilizza molte dll referenziate nei vari progetti, seguire il presente documento  per l’ordine di ricompilazione in caso di variazioni.

Il sito si compone di diversi frame inclusi uno nell’altro secondo la seguente struttura:

LOGO

CORPO

MENU

VISUALIZZA

COMANDI

VISUALIZZA

BASSO

NASCOSTO

GENERALE

I dati anagrafici vengono sempre esposti con la stessa pagina (Generali/asp/VisualAnag.aspx) che viene inclusa ove necessario tramite iframe.

In tutte le pagine html sono presenti i seguenti div per la gestione dell’elaborazione in corso e della messaggistica.

<div id="DivAttesa" runat="server" style="z-index: 101; position: absolute;display:none;">

<div class="Legend" style="margin-top:40px;">Elaborazione in Corso...</div>

<div class="BottoneClessidra">&nbsp;</div>

<div class="Legend">Attendere Prego</div>

</div>

<div id="divDialogBox" class="col-md-12">

<div class="modal-box">

<div id="divAlert" class="modal-alert">

<span class="closebtnalert" onclick="CloseAlert()">&times;</span>

<p id="pAlert">testo di esempio</p>

<input type="text" class="prompttxt"/>

<button id="btnDialogBoxeOK" class="confirmbtn Bottone BottoneOK" onclick="DialogConfirmOK();"></button>&nbsp;

<button id="btnDialogBoxeKO" class="confirmbtn Bottone BottoneKO" onclick="DialogConfirmKO();"></button>

<button id="CmdDialogConfirmOK" class="hidden" onclick="RaiseDialogConfirmOK();"></button>

<button id="CmdDialogConfirmKO" class="hidden" onclick="RaiseDialogConfirmKO();"></button>

<input type="hidden" id="hfCloseAlert" />

<input type="hidden" id="hfDialogOK" />

<input type="hidden" id="hfDialogKO" />

</div>

</div>

<input type="hidden" id="cmdHeight" value="0" />

</div>

In ogni caso fare riferimento al documento  per la metodologia di sviluppo, logging, versiong, rilascio e simili del progetto.

### Librerie

Col termine “librerie” qui si intende l’insieme di componenti riusabili e condivisi, a prescindere dal loro effettivo formato e dalla tecnologia impiegata per produrli. Fanno parte delle librerie anche i “files di procedura”, ossia l’insieme di routines specializzate richiamate da ciascun modulo applicativo e scritte nel medesimo linguaggio di programmazione usato per lo sviluppo delle applicazioni stesse.

### Base Dati.

È l’oggetto “contenitore” delle informazioni, ed è composto a livello fisico di più oggetti elementari.

Ad ogni modulo applicativo risponde un database.

# Funzioni applicative

Elenco di tutte le voci di menù implementate. Ogni singola voce può essere visibile o nascosta in base alle autorizzazioni dell’utente loggato e alle configurazioni dell’ente; stessa logica è applicata anche ai comandi all’interno delle videate che possono essere visibili o bloccati in base alle autorizzazioni dell’utente loggato, ma anche in base alle configurazioni dell’ente. In allegato i documenti di dettaglio delle funzioni.

# Problematiche

Ci sono alcune anomalie ricorrenti che non si riescono a replicare completamente o che sono state rimandate magari perché richiedono troppo tempo per l’analisi; nel documento allegato i dettagli.

# Banca dati

Nelle tabelle di gestione dichiarazioni e dovuto è registrato sia il dato in linea che il dato storico. Questa gestione viene fatta attraverso l’uso del campo DATA\_VARIAZIONE che se vuoto indica il dato in linea, se valorizzato indica il dato storico.

Visto che sullo stesso server possono essere installate più istanze del sistema per diversi clienti, i database avranno come suffisso \_<ambiente di installazione>.

I database e le tabelle utilizzati sono elencati nel documento allegato.