

Documento di riferimento architetturale per lo sviluppo applicativo

Lo sviluppo applicativo in Vittoria è fondamentalmente basato su tecnologia Microsoft .NET e SQL Server.

La mappa degli strumenti di sviluppo certificati per ogni specifico settore di sviluppo è la seguente:

Area	Sotto Area	Strumenti accettati	Strumenti presenti ma non più supportati
Applicativi LOB NewAge e NewAge Sinistri	Piattaforma web	ASP.NET MVC 5.0	ASP ASP.NET MVC 3.0/4.0 ASP.NET Web Forms
	Sviluppo client- side	Bootstrap (framework web responsive) Knockout js (Javascript MVVM framework) jQuery Kendo UI 2016	
	Framework per dashboard applicative	JDash (versione MVC)	
	Autenticazione	Custom Newage, SSO Oauth (in fase di valutazione)	
	Autorizzazione	Custom Newage	
	Servizi applicativi	WCF (binding BasicHttp) ASP .NET Web API	Web Service
	Logger applicativo	log4net	
	Configurazione applicativa	Database VittoriaConfiguration	Database Sinistri_config
	Inversione di controllo	Microsoft MEF (1.0)	
	Framework applicativo	.NET 4.6.1 .NET Core 1.0 (in fase di valutazione)	.NET 2.0/3.5/4.0
	Applicazioni batch	BatchLoader (custom Vittoria), BatchSinistri (custom Vittoria Sinistri)	Batch come singoli Assembly
	Modalità di accesso ai dati	ADO.NET, NHibernate 3.4, EF 4.0/5.0/6.0, Stored Procedure	HOP, Linq to SQL, EF 1.0, NHibernate 3.1
	Servizi di cache	Microsoft Windows Server AppFabric Redis (in fase di introduzione)	SharedCache
	Servizi di Workflow	Microsoft Sharepoint 2013, Microsoft WorkFlow Manager (in fase di valutazione)	Microsoft Sharepoint 2010
	Applicativi BPM	Appian	
	Motore di template per PDF	Gutenberg (c#), Crystal Reports .Net, GostScript	Java e OpenOffice, Crystal Reports
	Processi frequenti a timer	.NET WinService (NewEra Template)	
	Servizi di firma digitale	Custom basati su fornitore Unimatica	custom basati su DLL InfoCert
Applicazioni dispositivi in mobilità	Piattaforma web	Apache Cordova	Proprietari
Database	Motore relazionale	SQL Server 2008 R2 / 2014 Database noSql in corso di valutazione	
	Servizi di ETL	SSIS 2008 R2 DTSX	IBM Datastage 8.0

		SSIS 2014 DTSX in fase introduzione	
	Servizi di Workload automation	BMC Control-M	
Code/Messaggi	Server di messaggistica	Microsoft Message Queue (MSMQ) Message Broker Service (in fase di valutazione)	
	Processi di scodamento	.NET WinService (NewEra Template), WAS Activation (limitato a bassi volumi di chiamate)	
Reportistica / Business Intelligence	Report per BI Enterprise	SAP Business Object, SQL Reporting Services 2014	
	Dashboard Report e KPI	SAP BO Dashboards	
	Report per BI self service	PowerPivot 2010 su SQL Server 2012 e Sharepoint 2010 PowerPivot 2013 in fase di introduzione	
App esterne a NewAge	Piattaforma web	ASP.NET 4.6.1	ASP.NET 4.0 Sharepoint 2013
	Autenticazione	SSO Novell Directory Services	Custom
	Autorizzazione	Custom OAuth 2 in fase di valutazione	
CRM	Piattaforma	Microsoft Dynamics CRM 2015	

Regole architetturali

- Sviluppo della User interface in ASP.NET

Le applicazioni ASP.NET seguono il principio di Separation of Concern, quindi vengono utilizzati pattern per la scomposizione della responsabilità quali MVP, MVVM, ed MVC.

Per applicazioni ASP.NET seguiamo un pattern MVP Custom Vittoria, e recentemente stiamo istituendo un template con ASP.NET MVC.

La user interface non contiene mai riferimenti diretti ad accesso dati, a regole di business etc. Normalmente la logica inserita negli strati di user interface è relativa al rendering degli oggetti e al mantenimento dello stato applicativo.

Le applicazioni devono essere realizzate secondo un'ottica di sito responsivo, utilizzando come framework Bootstrap.

Le convenzioni grafiche sono definite in appositi file CSS centralizzati, dove viene definito un comportamento grafico comune per gli oggetti principali.

Le applicazioni pensate specificatamente per l'utilizzo in mobilità, dovranno rispettare le specifiche di corretta visualizzazione per i device supportati (in termini di tipologia, risoluzione e dimensioni schermo). L'elenco viene mantenuto e aggiornato dall'area IT-Strumenti e Standard.

- Sviluppo della logica di Business

La logica di business viene del tutto sviluppata all'interno di assembly separati e testabili. Il numero di questi assembly è normalmente tenuto volutamente basso. La separazione logica infatti è ottenuta grazie allo studio di namespace e cartelle favorite rispetto alla creazione di diversi assembly (la frammentazione in assembly aiuta la

separazione logica ma aumenta i tempi di build e la pericolosità di cadere in referenze circolari).

- Logging

Ogni classe sviluppata (in qualsiasi layer applicativo) deve possedere chiamate di log applicativo (viene usato log4net) sufficienti a permettere un troubleshooting di produzione efficace e veloce.

Vengono usati log applicativi gerarchici ai seguente livelli:

- DEBUG: informazioni di debug molto dettagliate, quasi mai presenti in produzione a meno di richieste specifiche
- INFO: informazioni sui processi, normalmente tracciate anche in produzione
- WARN: segnalazioni di anomalie pericolose ma non bloccanti per la produzione
- ERROR: segnalazioni di qualsiasi anomalia (vengono esposti tutti i dettagli)
- FATAL: segnalazioni di anomalie gravi che compromettono il sistema o l'integrità dei dati

Lo strumento di log usato è il file di testo, con una classe appositamente scritta per l'append di più thread allo stesso file.

- Configurazione

Per la configurazione applicativa IT-Vittoria ha normalizzato tutte le possibilità di configurazione al database. I parametri di configurazione delle applicazioni vivono all'interno di un database in cui ogni chiave ha un namespace e un valore. L'accesso a queste chiavi è fatto da una

classe di costanti generata dinamicamente tramite oggetti strongly typed per evitare errori a runtime.

Altre modalità di configurazione sono vietate.

- Sessione

La sessione viene gestita in modo custom tramite un componente realizzato ad hoc. Le modalità base di utilizzo del componente sono già previste all'interno degli oggetti base da cui ereditano le UI.

- Test automatici

Le classi di business che modellano, regole ed entità devono essere scritte rispettando il requisito di testabilità automatica.

IT-Vittoria non impone la scrittura di test unitari ad esempio con vincoli di code coverage minimo, anche se il loro uso viene consigliato, specialmente nei casi in cui non è possibile una verifica di funzionamento tramite test coded UI. Ai fini della validazione applicativa, vengono prediletti i test automatici guidati a singole specifiche funzionali, o test di integrazione. La qualità interna delle applicazioni è supportata da una batteria di test di non regressione di tipo coded UI (registrazione dei flussi UI e trasformazione dei flussi in unit test), soprattutto perché i flussi operativi della line of business di Vittoria incrociano diverse tecnologie molte volte obsolete (ASP Classic, COM, VB) difficilmente testabili con soli unit test.

Per favorire la testabilità automatica, ogni classe di business non inializza direttamente tutte le sue componenti ma viene utilizzato un motore di inversione di controllo inserito nel .NET 4.0, MEF.

Anche se MEF è stato costruito pensando ai modelli plugin a runtime, il core principale rimane comunque un ottimo framework di inversione di controllo.

Da un punto di vista architetturale, l'inversione di controllo è effettuata tramite iniezione delle dipendenze esterne con delle property (e non con costruttori o metodi).

- Processi sincroni e asincroni

IT-Vittoria modella le applicazioni del mondo assicurativo delle agenzie e la stragrande maggioranza delle funzioni viene chiamata all'interno di una web app e risolta in un ciclo di request e response.

Alcuni processi tuttavia sono così complessi o così time-consuming che la risoluzione non può avvenire in modo sincrono rispetto alla richiesta (pena la scalabilità) ma processata da sistemi di backend appositi.

Questi processi di natura asincrona vengono comunque modellati con classi .NET, stored procedure, DTSX o altre tecnologie e il meccanismo di accodamento di una richiesta e scodamento della coda di richieste avviene totalmente tramite Microsoft MSMQ 4.0 o altri sistemi di gestione messaggi come indicato nell'elenco delle tecnologie supportate.

Gli accodamenti avvengono da qualsiasi sistema (.NET, ASP, COM, etc....). Per lo scodamento viene utilizzato un template di .NET Windows Service per ogni coda specifica.

In tutti i casi in cui la tipologia delle operazioni lo consente, la gestione puntuale asincrona delle operazioni è fortemente incoraggiata rispetto alla gestione massiva via batch.

- Servizi web

IT-Vittoria modella software per le proprie agenzie ma effettua servizi web anche per fornitori. I servizi sono modellati in .NET in modalità ASMX e WCF ed esposti con protocollo WS-* BasicProfile 1.1 e SOAP 1.2, in questo modo viene garantita l'interoperabilità attraverso altre tecnologie.

I servizi web vengono usati anche all'interno della nostra factory per superare eventuali incompatibilità tra applicazioni interne (framework differenti) o tecnologie differenti (COM, ASP e .NET).

Sono inoltre utilizzati servizi ASP.NET Web Api per snellire le comunicazioni relative ad app Mobile e applicazioni web pensate in ottica responsive.

