Software Architecture Document

Version <1.0>

Applicativi Touch in ambiente Microsoft- 2017

Daniele Zuppello

Indice

Sommario

[1. Introduzione 3](#_Toc491793820)

[1.1 Obiettivo 3](#_Toc491793821)

[1.2 Definizioni, Acronimi, e Abbreviazioni 3](#_Toc491793822)

[1.3 Riferimenti 3](#_Toc491793823)

[2. Rappresentazione architetturale 4](#_Toc491793824)

[3. Fattori architetturali 4](#_Toc491793825)

[3.1 Piattaforme 4](#_Toc491793826)

[4. Decisioni architetturali 5](#_Toc491793827)

[4.1 Promemoria tecnico <Password Hashing> 5](#_Toc491793828)

[5. Logical View 5](#_Toc491793829)

[5.1 Overview 5](#_Toc491793830)

[5.2 Package significativi 5](#_Toc491793831)

[6. Deployment View 5](#_Toc491793832)

[7. Use-Case View 6](#_Toc491793833)

[7.1 Realizzazioni di Use-Case 6](#_Toc491793834)

[8. Implementation View 6](#_Toc491793835)

[9. Data View 6](#_Toc491793836)

[10. Size and Performance 6](#_Toc491793837)

[11. Quality 6](#_Toc491793838)

# Introduzione

Il sistema operativo Windows non è più ormai un’esclusiva dei computer che potremmo definire “classici”. E’ ormai dalla versione 8 che esso è approdato anche su tablet, smartphone ed in generale su device che richiedono, od hanno la possibilità, di essere usati tramite uno schermo touch. Questo documento si inserisce in una attività di studio per cercare di comprendere come sviluppare applicativi touch in ambienti Microsoft

## Obiettivo

L’obiettivo dato dallo Stackholder è quello di riuscire a creare degli applicativi touch utilizzabili in parallelo da più persone.

Questi applicativi dovranno essere formati da una serie di widget con cui l’utente potrà interagire muovendoli e ruotandoli sullo schermo.

Il Widget sarà quindi l’unità base del sistema attraverso la quale l’utente interagirà con lo stesso

## Definizioni, Acronimi, e Abbreviazioni

|  |  |
| --- | --- |
| Widget | Unità base di interazione tra l’utente e l’applicativo. Fa parte della UI |
| Sistema o Applicativo | Con il termine "sistema" si indica il software, oggetto di questo documento. |
| Stakeholder | Lo stakeholder è il committente del sistema. |
| UWP | Universal Windows Platform, ovvero una tipologia di applicativi utilizzabili su qualsiasi device con installato un sistema operativo Windows 10 |
| UI | User Interface, l’interfaccia utente |
|  |  |
|  |  |
|  |  |

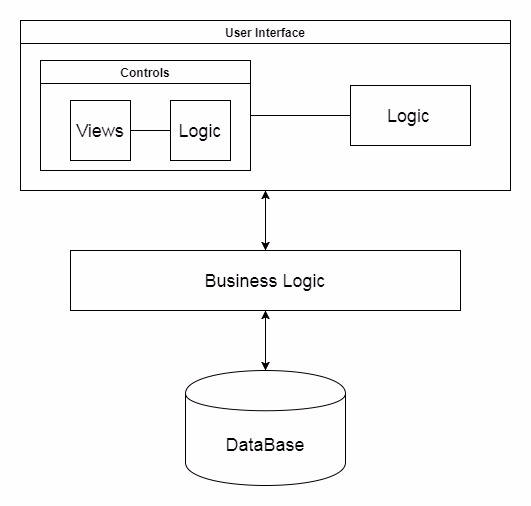
## Riferimenti

Documento SRS (sempre dello stesso progetto) v 1.0

<https://docs.microsoft.com/en-us/windows/uwp> (Documentazione ufficiale Microsoft)

<https://github.com/Microsoft/Windows-universal-samples> (Esempi ufficiali Microsoft)

# Rappresentazione architetturale



# Fattori architetturali

Lo stackeholder ha espressamente richiesto che il sistema fosse sviluppato pensando a quanttro precisi fattori architetturali:

1. Flessibilità
2. Semplicità di sviluppo
3. Riusabilità del codice
4. Manutenibilità del codice

Si è inoltre cercato di mantenere l’interfaccia utente il più semplice possibile per rendere l’applicativo di facile utilizzo.

## Piattaforme

L’applicazione è implementata con il linguaggio di programmazione C# versione e con il framework .NET

Essendo l’applicativo di tipologia UWP potrà essere utilizzato su qualsiasi device su cui è installato Windows 10.

# Decisioni architetturali

Per ottenere un grado di manutenibilità soddisfacente è necessario separare il più possibile parti di codice che implementano funzionalità differenti. È dunque opportuno dividere l'accesso ai dati, la logica di business, e la presentazione delle informazioni.

Tale separazione rende anche il codice facilmente riutilizzabile in quanto si riescono a creare porzioni di codice che non dipendono dall’applicativo in sé ma che devono adempiere solo ad un determinato compito (quale può essere l’accesso ad un database).

Anche la flessibilità e la semplicità di sviluppo sono accentuate dalla suddetta divisione poiché si può andare a modificare una determinata funzionalità di uno strato senza andare ad impattare sugli altri strati (a patto, ovviamente, di non cambiare le firme dei metodi che si vanno a modificare ma solo le funzionalità). Ciò permette quindi di dividere il lavoro con più facilità e quindi di poter far lavorare in parallelo più persone riuscendo anche a simulare facilmente un risultato aspettato di input o output

# Logical View

## Overview

## Package significativi

# Deployment View

# Use-Case View

## Realizzazioni di Use-Case

# Implementation View

# Data View

# Size and Performance

# Quality