



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

TESI DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

# APPLICAZIONE DELLA TECNOLOGIA RFID AI BENI CULTURALI

RELATORE: Lorenzo Vangelista

LAUREANDO: Simone Zennaro

24 Settembre 2012

Anno Accademico 2011/2012



*Alla mia famiglia che in questi anni mi ha sempre sostenuto.*

*Un ringraziamento di tutto cuore alla mia splendida ragazza, Virginia, che mi ha sempre aiutato e sostenuto nei momenti difficili.*



# INDICE

Sommario .....	7
1 La mostra: "I segreti della città proibita. Matteo Ricci alla corte dei Ming .....	9
2 La tecnologia rfid .....	11
2.1 Che cos'è RFID .....	11
2.2 Classificazione dei sistemi RFID .....	13
2.2.1 Classificazione per fonte d'alimentazione .....	13
2.2.2 Classificazione per elaborazione dei dati .....	13
2.2.3 Classificazione per tipo di memoria .....	14
2.2.4 Classificazione per frequenza di lavoro .....	15
2.3 Standard e Normative .....	16
2.4 Protocolli ISO-15693 e ISO-14443 .....	17
2.4.1 ISO-14443 .....	17
2.5 MiFare .....	19
2.5.1 Classic .....	19
2.5.2 Plus .....	20
2.6 Il lettore utilizzato .....	20
2.7 APDU .....	21
3 Analisi delle specifiche .....	23
3.1 Perché RFID .....	23
3.2 Obiettivi .....	24
3.3 Requisiti .....	25
4 Tecnologie e infrastrutture del sistema .....	27
4.1 MySQL .....	27
4.2 C# .....	28
4.3 Flash .....	30
4.5 LogMeIn .....	31
5 Il programma .....	33
5.1 Client .....	33
5.2 Server .....	37
5.3 installazione del sistema .....	39
6 Valutazione e testing del sistema .....	41
6.1 Affidabilità e solidità del sistema .....	41
6.2 Problemi riscontrati .....	41
7 Conclusioni .....	43
Tavola degli acronimi .....	45
Appendice .....	47
Bibliografia .....	49



## SOMMARIO

In questa tesi si analizza lo sviluppo di un sistema basato sulla tecnologia RFID e di come sia stata utilizzata all'interno di una mostra. In particolare tratteremo del software realizzato per intrattenere gli ospiti di tale mostra e di come la tecnologia RFID sia stata utile. Il progetto è stato realizzato da una ditta con cui si è collaborato. L'intero ecosistema, che è stato installato, quindi non può essere solo frutto del lavoro di cui si tratterà, anche se esso ne è una parte fondamentale.

Si comincerà dalla descrizione della tecnologia RFID per proseguire con una discussione sulle specifiche del sistema. Poi saranno esposte informazioni sulle altre tecnologie usate e infine si parlerà del programma realizzato.

2. **La tecnologia RFID:** s'inizierà parlando della tecnologia RFID; descrivendone le proprietà e le potenzialità. Il capitolo si concluderà con una breve descrizione del lettore utilizzato.

3. **Analisi delle specifiche:** seguirà un'analisi dell'ambiente in cui sarà installato il programma discutendo degli obiettivi del sistema e dei suoi requisiti.

4. **Tecnologie e infrastruttura del sistema:** tratterà delle altre tecnologie implementate nel sistema e del perché siano state scelte.

5. **Il programma:** è illustrato il funzionamento del programma realizzato, sia della parte lato Client sia di quella lato Server.

6. **Valutazioni e testing del sistema:** l'ultimo capitolo tratterà dei test effettuati sul sistema a realizzazione conclusa e dei problemi che riscontrati nella realizzazione del software.

Seguiranno le conclusioni, dove si riporteranno dati finali e valutazioni conclusive.





## 1 LA MOSTRA: “I SEGRETI DELLA CITTÀ PROIBITA. MATTEO RICCI ALLA CORTE DEI MING”



La mostra, “*I segreti della Città Proibita. Matteo Ricci alla corte dei Ming*”, è stata ospitata dal 24 ottobre 2009 fino al 9 maggio 2010 a Treviso dove, presso la Casa dei Carraresi, sono stati esposti **300 reperti**, tra gioielli, porcellane, statue d’oro e mobili della Cina. Questa esposizione è il naturale proseguimento del percorso iniziato, nel 2005, con “La nascita del celeste Impero” e, nel 2007, con “Gengis Khan e il Tesoro dei Mongoli”, tutte mostre, comunque, sempre curate e organizzate dalla Fondazione Cassamarca.

*La Città Proibita*, sopra citata, fa riferimento alla favolosa reggia di Pechino fatta costruire tra il 1406 e il 1421 da Yongle, terzo imperatore della dinastia dei Ming ed è la protagonista di ciclo espositivo che si snoda attorno e dentro la straordinaria città imperiale che occupa il centro di Pechino.

Per l’occasione, la Casa dei Carraresi è stata arricchita dagli allestimenti tecnologici e interattivi realizzati da **Fabrica**, il centro di ricerca sulla comunicazione del gruppo Benetton in collaborazione con EasyPC. Grazie al computer, la mostra è percorribile da tutti, ma soprattutto dai bambini, attraverso un percorso tecnologico che permette di addentrarsi virtualmente all’interno della Città Proibita. Il software che ha fatto da scheletro per le applicazioni, che successivamente saranno illustrate, è stato pensato e realizzato al fine di offrire la migliore esperienza agli utenti.

Inoltre, una sezione speciale è stata dedicata a **Matteo Ricci**, padre gesuita, matematico e astronomo che intraprese un’azione missionaria nel territorio cinese al tempo della Dinastia dei Ming; tale figura religiosa è stata presentata attraverso documenti, testi e meccanismi di cui si serviva per la sua attività di astronomo.



## 2 LA TECNOLOGIA RFID

### 2.1 CHE COS'È RFID

L'**RFID** (o Radio Frequency Identification o Identificazione a radio frequenza) è una tecnologia per l'identificazione e/o memorizzazione dati. Con essa si possono memorizzare informazioni di vario genere, ad esempio di oggetti, animali o persone (*AIDC Automatic Identifying and Data Capture*) utilizzando particolari dispositivi elettronici (detti *tag* o *transponder*), capaci di memorizzare dati. Questi tag sono in grado di rispondere all'interrogazione a distanza da parte di appositi apparati, fissi o portatili, chiamati per semplicità "lettori" a radiofrequenza, comunicando (o aggiornando) le informazioni in essi contenute. In un certo senso possono essere quindi assimilabili a sistemi di "lettura e/o scrittura" senza fili, con numerosissime applicazioni.

Sebbene lo sviluppo di questa tecnologia si sia avviato solo da metà degli anni '90, l'RFID nacque con l'invenzione dei radar, avvenuta durante la seconda guerra mondiale. In particolare, l'applicazione di questi dispositivi innovativi è dovuta alla necessità da parte della Gran Bretagna di trovare un modo per distinguere i propri aerei da quelli nemici: venne così abbandonato il radar, che consentiva solamente l'individuazione di aerei, senza dare informazioni sulla loro nazionalità.

Si ebbe, quindi, la necessità di costruire un sistema detto "identification friend or foe", cioè un sistema grazie al quale era possibile l'identificazione dell'oggetto: se amico o nemico.

Si decise, pertanto, di implementare sui velivoli inglesi e alleati una scatola contenente una ricetrasmittente, denominata in seguito "transponder", che, all'atto dell'illuminazione radar, rispondesse sulla stessa frequenza con un "bip", permettendo al radar stesso l'identificazione dei velivoli. Il passo successivo fu quello di creare un'identificazione univoca del velivolo, mediante un ID; ciò fu possibile modulando l'emissione del transponder a bordo dell'aereo, che quindi non inviava più un semplice impulso, ma una serie d'impulsi opportunamente codificati. Questa innovazione permise di "numerare" gli aerei e conoscerne la posizione univoca.

Questo fu lo sviluppo iniziale della tecnologia RFID soltanto nell'ambito militare. Dalla metà degli anni '90 in poi, invece, l'RFID cominciò a diffondersi negli aeroporti, nella gestione delle merci e nelle catene di produzione, utilizzando un sistema chiamato EAS (Electronic Article Surveillance), che consentiva unicamente la verifica che uno specifico tag fosse presente in un certo campo d'azione molto limitato.

Lo sviluppo e la diffusione dell'RFID è diretta conseguenza della miniaturizzazione e standardizzazione mediante modello OSI (Open Systems Interconnection) dei circuiti elettronici.



FIGURA 1 - EVOLUZIONE DELLA TECNOLOGIA RFID

Un sistema RFID è composto di quattro elementi base:

- Un'antenna;
- Un lettore, comunemente chiamato Reader;
- Un trasponder, chiamato Tag, in grado di contenere dati;
- Un middleware, un insieme di software, presente nel PC, in grado di elaborare e interpretare i dati ricevuti.

Se il sistema ha bisogno della connessione a un server applicativo, sarà necessaria una rete.

L'antenna e il trasponder comunicano tra loro attraverso un segnale modulato a radiofrequenza. L'antenna del lettore genera un campo elettromagnetico a bassa potenza all'interno del quale il PICC (Proximity Card) è in grado di funzionare; poi i dati ricevuti dal lettore saranno inviati a un PC che li elaborerà secondo necessità.

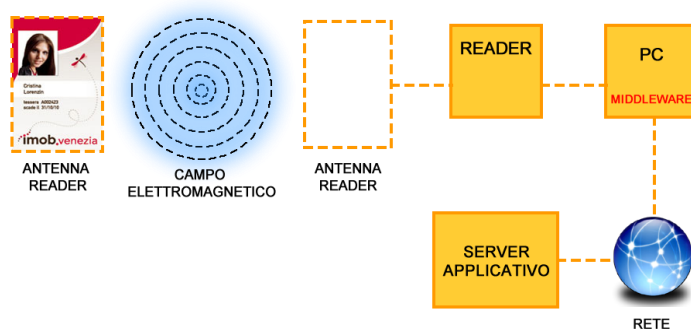


FIGURA 2 - SCHEMA DI UN SISTEMA RFID

Importanti caratteristiche dei tag sono:

- Frequenza di lavoro
- Distanza tipica di lavoro
- Aderenza ai vari standard
- Quantità di memoria
- Tipo di memoria
- Velocità d'identificazione
- Velocità di trasmissione

## 2.2 CLASSIFICAZIONE DEI SISTEMI RFID

Un sistema RFID può essere classificato per vari parametri: frequenza di lavoro, tipo di memoria utilizzata, alimentazione e modalità di lavoro sono solo alcune di queste.

### 2.2.1 Classificazione per fonte d'alimentazione

I sistemi RFID, nella classificazione per fonte d'alimentazione, possono essere divisi in: attivi, passivi e semipassivi.

Nei sistemi passivi il trasponder recupera l'energia, necessaria al suo funzionamento, dal segnale che è inviato dal reader, che è un elemento attivo. Il tag, di dimensioni ridotte e costi contenuti, quindi, è utilizzabile solamente se esposto ad esso.

I sistemi semipassivi, invece, presentano la parte logica alimentata da una batteria mentre la parte di comunicazione è passiva. Essendo la parte logica alimentata da una piccola batteria, può permettersi una complessità circuitale maggiore e quindi implementare funzionalità più evolute e versatili.

Infine, nei sistemi attivi, di cui il TELEPASS è un rappresentante, sia la parte logica sia quella di trasmissione sono alimentate da una batteria, aumentando così la portata; di conseguenza i costi di questa soluzione sono più elevati (circa 50-100 volte maggiori ai sistemi passivi).

### 2.2.2 Classificazione per elaborazione dei dati

Si può classificare il tag in base all'elaborazione dei dati e alla capienza di memoria, si possono così individuare tre tipologie di sistema: low-end, mid-range e high-end.

I sistemi EAS (Electronic Article Surveillance), utili in alternativa del codice a barre, rappresentano l'estremo inferiore dei sistemi di tipo low-end. Questi verificano e controllano la possibile presenza di un tag nella zona d'interrogazione del lettore, basandosi semplicemente su di una risonanza LC a una prefissata frequenza. I dati in essi contenuti, spesso composti esclusivamente da un numero di serie, sono memorizzati all'interno di una ROM (Read Only Memory) e l'invio degli stessi è subordinato all'ingresso del Tag nell'area attiva. La comunicazione è quindi unilaterale: solo il Tag può comunicare con il lettore.

I sistemi RFID con tag a memoria riscrivibile sono classificati invece come sistemi mid-range; la loro capacità di memoria varia tra alcuni byte a svariati kbyte sia per memorie EEPROM (Electrically Erasable Programmable Read-Only Memory) che SRAM (Static Random Access Memory) e prevedono anche un protocollo di gestione delle collisioni, in grado di gestire la presenza di più Tag nell'area attiva.

Della categoria dei sistemi high-end fanno parte Tag con un microprocessore e un sistema operativo interno che li abilita a svolgere procedure complesse e a proteggere i dati in maniera molto efficace.

### 2.2.3 Classificazione per tipo di memoria

Un'altra caratteristica che differenzia i transponder sono le memorie di cui sono dotati; si possono raggruppare per memorie di sola lettura (ROM e WORM, acronimo di Write Once Read Many) o riscrivibili (RAM – Random Access Memory –, EEPROM, FRAM – Ferroelectric Ram –).

Le ROM sono configurate dal produttore dell'elemento con un numero limitato d'informazioni e l'uso risulta poco flessibile.

Le WORM permettono invece di scriverla una volta sola, senza poterne in seguito cancellare il contenuto, permettendo all'utente di personalizzare il tag.

La memoria RAM permette di raggiungere grandi densità di dati memorizzati a costi competitivi; necessita tuttavia di una fonte d'energia costantemente attiva per mantenere i dati e, quindi, si presta a un utilizzo solo su Tag attivi.

Le memorie EEPROM hanno il vantaggio di richiedere la presenza di una fonte d'energia soltanto durante l'operazione di lettura/scrittura in memoria e possono contenere i dati al loro interno anche senza alimentazione per anni; questo le classifica come miglior compromesso per tag passivi.

Talvolta, a causa dell'elevato tempo di scrittura, la tecnologia EEPROM è abbandonata in favore della FRAM che, grazie alla polarizzazione elettrica della memoria, è in pratica esente da disturbi da campi elettromagnetici estranei; il tempo di scrittura su tale memoria è fino a dieci volte minore rispetto all'EEPROM.

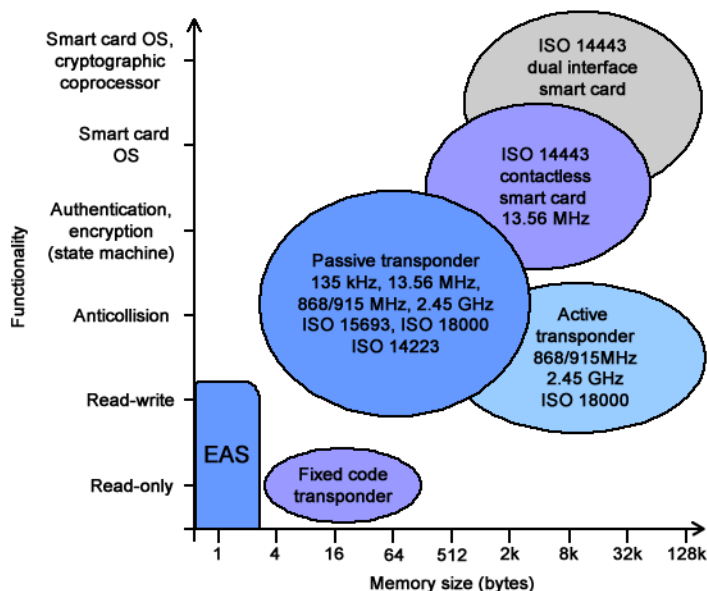


FIGURA 3 - CLASSIFICAZIONE PER DIMENSIONE DELLA MEMORIA

## 2.2.4 Classificazione per frequenza di lavoro

Si possono classificare le bande di frequenza utilizzate nei sistemi RFID in:

• LF (Low Frequency)	} Accoppiamento Magnetico (induttivo)
• MF (Medium Frequency)	
• HF (High Frequency)	
• UHF (Ultra High Frequency)	} Accoppiamento Elettromagnetico
• Microwaves, per frequenze superiori a 2 GHz.	

	Frequenza	Applicazione	Note
Il BitRate aumenta	125(*) kHz 134kHz	<ul style="list-style-type: none"> <li>Controllo accessi, controllo animali</li> <li>Sistemi antifurto di bloccaggio delle automobili</li> </ul>	<ul style="list-style-type: none"> <li>Velocità di trasmissione 10kbps</li> <li>Interferenze dai monitor delle installazioni industriali</li> </ul>
	195 3.25 MHz 475/8.2 MHz	<ul style="list-style-type: none"> <li>Etichette EAS nei depositi di vendita al minuto</li> </ul>	
	13.56(*) MHz (Banda ISM)	<ul style="list-style-type: none"> <li>Trasponders read only, read/write ed EAS</li> </ul>	<ul style="list-style-type: none"> <li>Velocità di trasmissione 100kbps</li> <li>Interferenze da altri utenti della banda ISM entro un raggio di qualche metro</li> </ul>
	420-460 MHz 869 MHz	<ul style="list-style-type: none"> <li>Identificazione di veicoli (ferrovie)</li> <li>Identificazione di oggetti in movimento</li> </ul>	
	2.35-2.45(*) GHz	<ul style="list-style-type: none"> <li>Tag generalmente attivi</li> </ul>	
	5.4-5.9 GHz	<ul style="list-style-type: none"> <li>Controllo veicolare e applicazioni future</li> </ul>	
	24.125 GHz	<ul style="list-style-type: none"> <li>Sistemi di allarme</li> <li>Misuratori di velocità</li> <li>Aperture automatiche</li> </ul>	

TABELLA 1 - CLASSIFICAZIONE DEI SISTEMI RFID PER FREQUENZA DI LAVORO

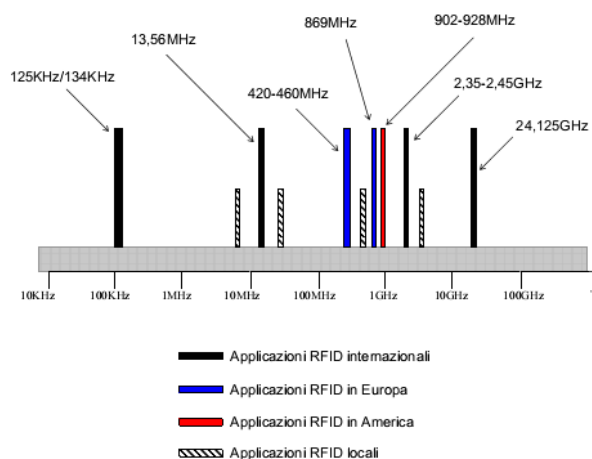


FIGURA 4 - FREQUENZE DI LAVORO

## 2.3 STANDARD E NORMATIVE

Vista la grande gamma di frequenze e tag è ovvio pensare che non esista un lettore in grado di leggere qualsiasi tipo di tag. L'EPC Global2, azienda che porta avanti la normativa sui sistemi RFID, con il termine "Agile Reader" identifica quei lettori in grado di leggere diversi tipi di tag, operando anche a differenti frequenze, sempre però all'interno della banda di lavoro.

I lettori devono rispettare rigorosamente le normative internazionali che regolano le trasmissioni radio per ogni paese, ed è questo uno dei più grossi problemi della diffusione dei sistemi RFID.

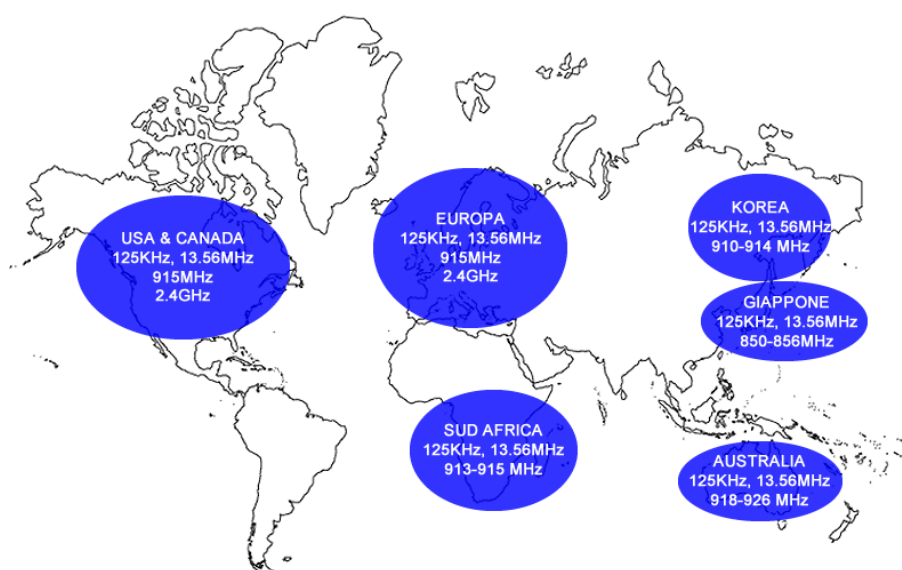


FIGURA 5 - FREQUENZE NEL MONDO



## 2.4 *PROTOCOLLI ISO-15693 E ISO-14443*

I protocolli per sistemi RFID a 13.56 MHz sono due e, assieme, regolano le caratteristiche che dovrebbero avere le smartcard operanti senza contatto diretto, le così dette contactless card. Sebbene l'appartenenza a un protocollo comune abiliti due lettori alla comunicazione è libera scelta del produttore decidere se integrare o no tale tecnologia all'interno dei suoi prodotti.

Il Protocollo ISO-15693 regola la comunicazione delle smartcard, dette Vicinity Cards, operanti a una distanza maggiore rispetto alle Proximity Cards e, precisamente, fino a una distanza di 1, 1.5 metri.

Lo standard ISO-14443 è anche detto ISO/IEC-14443 perché è stato sviluppato dal comitato tecnico congiunto ISO e IEC. ISO (International Organization for Standardization) è l'Organizzazione internazionale per la normazione fondata il 23 febbraio 1947 ed ha il suo quartier generale a Ginevra in Svizzera. IEC (International Electrotechnical Commission) è un'organizzazione internazionale per la definizione di standard in materia di elettricità, elettronica e tecnologie correlate; è stata fondata nel 1906 e ha sede a Ginevra anch'essa.

### 2.4.1 **ISO-14443**

ISO-14443 definisce lo standard per le contactless smartcard, appartenenti alla famiglia delle Proximity Card, che lavorano a una distanza massima di 10 cm. Sono tessere in genere più potenti delle Vicinity Cards e che, per questo, richiedono maggiore energia. Ne esistono alcuni tipi con veri processori general purpose che si avvalgono di Sistemi Operativi integrati. Possono anche essere dotate di sistemi di sicurezza e crittografia dei dati più o meno potenti e affidabili.

In questo genere di tessere esistono due tipi di dato: R/W Block, per immagazzinare 16 byte di dati generalizzati, e il Value-Block, usato per operazioni consecutive d'incremento, decremento e ripristino di valori. La dimensione massima di un Value-Block è di 4 byte, ma occupa lo stesso un intero blocco da 16 byte, replicando 3 volte il valore per preservarlo da errori, e riservando, in parte, 4 byte come bit di controllo. Un dato di tipo Value-Block può risiedere in qualsiasi blocco di memoria ma va in precedenza inizializzato.

Il protocollo si divide ed è descritto in 4 parti:

1. Caratteristiche Fisiche;
2. Radiofrequenza e Modulazione;
  - a. Modalità A;
  - b. Modalità B;
3. Inizializzazione e Anticollisione;
4. Protocollo di Trasmissione.

Nel protocollo è prevista una sezione apposita per regolamentare la gestione di Tag multipli nell'area attiva e le potenziali collisioni che ne seguono, sebbene le dimensioni ridotte del lettore e la vicinanza richiesta per il funzionamento ne limitino la presenza.

Per la comunicazione sono disponibili più modalità:

- Addressed Mode (ADD): ad ogni comando viene associato l'UID del Tag che deve eseguire il comando;
- Selected Mode (SEL): prima viene selezionato un Tag e poi ogni comando è destinato a quel Tag;
- Non-Addressed Mode (NAD): il comando è mandato in Broadcast a tutti i Tag nell'area attiva.

I comandi che possono essere inviati al Tag sono riassunti nella tabella seguente.

Comando	Modalità	Descrizione
Inventory	NAD	Controlla l'esistenza di Tag nell'area attiva e restituisce alcuni dati, ES: "UID".
Select	ADD	Seleziona un Tag.
Authentication	SEL	Autenticazione tramite chiave.
Halt	SEL	Il tag selezionato è messo nello stato di HALT.
Read multiple Blocks	ADD,SEL	Legge una serie di blocchi dalla carta (16 byte l'uno).
Write multiple Blocks	ADD,SEL	Scrive una serie di blocchi di dati sulla carta (16 byte l'uno).
InitValue	SEL	Inizializza un blocco di memoria come Value-Block.
ReadValue	SEL	Legge il valore di un Value-Block.
Increment	SEL	Somma il valore passato con quello di un Value-Block e salva il risultato nel registro interno.
Decrement	SEL	
Restore	SEL	Legge un Value-Block, controlla che non ci siano errori e salva il valore nel registro interno.
Transfer	SEL	Trasferisce il valore del registro interno al Value-Block.

TABELLA 2 – LISTA DEI COMANDI ISO-14443

## 2.5 MIFARE

MiFare è la tecnologia di contactless smartcard più diffusa al mondo. La tecnologia proprietaria MiFare, brevettata da NXP Semiconductors, è basata sullo standard ISO 14443, tipo A, ed è inserita sia nelle smart card sia nei lettori.

Il nome MiFare si riferisce a due diversi tipi di carte contactless:

- La carta MiFare Standard (o Classic) impiega un protocollo proprietario Philips per l'autenticazione e l'elaborazione. Le carte MIFARE Ultralight impiegano lo stesso protocollo, ma senza lo standard di sicurezza.
- La carta MiFare Plus è compatibile con la versione standard ma adotta un migliore sistema di sicurezza.

### 2.5.1 Classic

Le carte MiFare Classic sono implementate con circuiti integrati ed hanno una ridotta potenza di calcolo; ciò le porta a rappresentare un semplice dispositivo di memoria dove lo spazio è diviso in segmenti e blocchi, protetti da meccanismi di controllo degli accessi; ne esistono di più dimensioni: le MiFare Classic 1K, le MiFare Classic 4K.

Le MiFare Classic 1K dispongono di 1024 byte di memoria suddivisi in 16 segmenti; ogni settore è protetto da 2 chiavi che vengono chiamate A e B. In tutti i tipi di carta, 16 byte a settore sono riservati alle chiavi e alle condizioni d'accesso e non possono essere utilizzati per i dati dell'utente; inoltre, i primi 16 byte della tessera contengono il numero di serie univoco e di sola lettura. In questo modo la memoria disponibile si riduce a 752 byte. Le MiFare Classic 4 K dispongono invece di 4096 byte, divisi in 40 segmenti, di cui 32 sono delle stesse dimensioni dell'1K mentre i restanti 8 hanno dimensioni quadruple. Le dimensioni della memoria disponibile si riducono, per le ragioni di cui sopra, a 3440 byte.

In queste carte sono rispettate solo le prime tre parti del protocollo ISO-14443 (caratteristiche fisiche, potenza e interfaccia del segnale radio, inizializzazione e anticollisione), mentre un protocollo proprietario sostituisce la quarta parte: trasmissione dati. La presenza di un protocollo proprietario impedisce l'accesso tramite lettori e funzioni standard del protocollo ISO-14443.

Queste tessere sono le più diffuse e utilizzate grazie al basso costo e all'affidabilità che garantiscono; implementano l'algoritmo di crittografia Crypto-1 creato da NXP semiconductors.

## 2.5.2 Plus

MiFare Plus nasce per sostituire il modello “classico” e per questo ne condivide la gestione dei dati e delle applicazioni; fornisce solo un aggiornato e migliorato sistema di sicurezza che però rende inadatti i lettori per MiFare Classic e ne obbliga l’aggiornamento. Queste tessere supportano AES (Advanced Encryption Standard) a 128 bit il che le rende molto più sicure e affidabili.

## 2.6 IL LETTORE UTILIZZATO



Il lettore utilizzato è prodotto dall’ACS - Advanced Card Systems Ltd con sede a Hong Kong. Il modello in questione è l’ACR122U, una smart card reader/writer che si collega via USB; in grado di accedere a ISO 14443-4 tipo A e tipo B, Mifare e NFC (Near Field Communication).

Per lo sviluppo delle applicazioni, il lettore è venduto con un CD contenente l’SDK (Software Development Kit) che si compone di esempi pratici e codici già pronti nei principali linguaggi di programmazione.

Per la comunicazione tra il lettore e il computer sono utilizzate delle librerie dinamiche, comunemente dette DLL, che sono caricate quando si esegue il programma, runtime. In particolare si usa una libreria sviluppata dalla Microsoft, già presente nel sistema operativo Windows, il cui nome è: “winscard.dll”. Quest’aspetto torna utile al programmatore che, grazie all’utilizzo di strumenti standard, riesce a reperire facilmente documentazione e materiale per lo sviluppo. L’accesso tramite questa libreria si colloca a un livello di astrazione medio-basso permettendo un accesso efficiente alle risorse a patto di conoscere bene il sistema con cui si ha a che fare, sia da un punto di vista software che hardware.

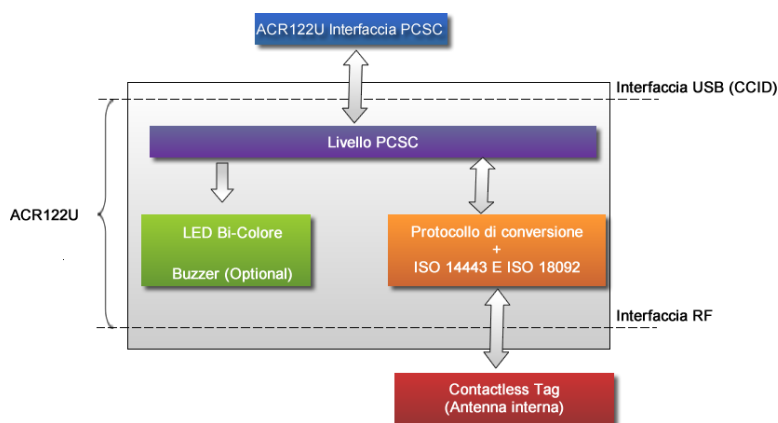


FIGURA 6 - ARCHITETTURA DEL LETTORE

## 2.7 APDU

La comunicazione delle informazioni tra la tessera e la smart card avviene attraverso lo scambio di pacchetti detti APDU (Application Protocol Data Unit); essi rappresentano l'unità di comunicazione tra il lettore e la tessera e la loro struttura è definita da "ISO/IEC 7816-4 Organization, security and commands for interchange".

Ci sono due tipi di APDU: quelli di comando e quelli di risposta. I primi presentano dei campi attraverso i quali il lettore sa che operazioni compiere e quali dati inviare, i secondi contengono il risultato dell'operazione richiesta.

La struttura dei comandi è qui riassunta in delle tabelle.

Comandi APDU		
Nome del campo	Lunghezza (bytes)	Descrizione
CLA	1	Classe dell'istruzione – indica il tipo di comando.
INS	1	Codice dell'istruzione – indica il comando specifico, ES: "Leggi dati".
P1-P2	2	Parametri dell'istruzione, ES: "L'offset nel file in cui scrivere i dati".
Le	0,1,2 o 3	Il numero ( $N_e$ ) massimo di bytes nella risposta

TABELLA 3 - STRUTTURA DEI COMANDI APDU

Risposta APDU		
Nome del campo	Lunghezza (bytes)	Descrizione
Dati	Da 0 a $N_r$	Dati di risposta.
SW1-SW2	2	Risultato del comando, ES: "90 00 (esadecimale) indica il successo".

TABELLA 4 - STRUTTURA DELLE RISPOSTE APDU

Ora che è più chiaro il significato dei termini vediamo i parametri specifici da usare per comunicare con il lettore che stiamo usando.

Comando	Classe	INS	P1	P2	Le
Ottenere Dati	FF	CA	00	00	00 (Lunghezza Totale)

TABELLA 5 – RICHIESTA APDU PER OTTENERE L'UID DEL PICC

La risposta ottenuta è strutturata come evidenziato dalla tabella 6 e ha una lunghezza pari a quella dei dati più 2 byte; l'UID (User Identifier) è restituito con il bit meno significativo a sinistra (quindi usa il sistema little-endian) ed è seguito dai campi che segnalano l'esito del comando.

<b>Risposta</b>	<b>Dati</b>					
Risultato	UID (LSB)			UID (MSB)	SW1	SW2

TABELLA 6 - RISPOSTA APDU CONTENENTE L'UID

Confrontando i valori dei bytes SW1 e SW2 con quelli riportati in tabella 8 è possibile evincere se il comando è andato a buon fine.

<b>Risultato</b>	<b>SW1</b>	<b>SW2</b>	<b>Significato</b>
Successo	90	00	L'operazione è stata conclusa con successo.
Errore	63	00	Operazione fallita
Errore	6A	81	Funzione non supportata

TABELLA 7 - CODICI DI RISPOSTA

## 3 ANALISI DELLE SPECIFICHE

### 3.1 PERCHÉ RFID

La necessità di identificare in modo semplice e veloce gli utenti hanno spinto a pensare a varie soluzioni. Subito quando si pensa all'identificazione viene spontaneo pensare a codici univoci e magari ai codici a barre che si è abituati a conoscere andando per negozi. L'utilizzo di questi ultimi non poteva rappresentare la soluzione cercata dato che il pubblico d'interesse dell'applicazione era principalmente composto di bambini. Sperare che questi si prestino a passare in modo corretto il codice davanti ad un lettore era quanto meno illusorio; è esperienza comune che il codice a barre non sempre viene rilevato correttamente e spesso sono necessari più tentativi e pazienza affinché venga riconosciuto correttamente.

La tecnologia RFID presenta tutti i vantaggi dell'identificazione univoca del codice a barre unita alla semplicità nell'utilizzo. Basta, infatti, accostare la tessera al lettore in qualsiasi modo o verso perché questa funzioni. Le tessere RFID, dette transponder o TAG, si possono trovare in commercio sotto varie forme: dalla tessera, all'adesivo e sono facilmente personalizzabili. Altro aspetto molto importante nella scelta è stato il costo; i TAG si trovano a cifre irrisorie e i lettori sono generalmente reperibili a costi accettabili e quindi la scelta, non solo era funzionale tecnicamente, ma anche economicamente. Se tutte le ragioni fin qui esposte non fossero convincenti, basta ricordare che queste tessere hanno anche una memoria interna che ben si presta a implementazioni future più complesse. Si potrebbe anche pensare di creare una serie di mostre il cui biglietto d'ingresso è costituito dalla tessera RFID e con la quale è possibile accedere in luoghi diversi; portando eventuali punteggi o informazioni, riguardanti l'attività precedentemente svolta, sempre con se.



FIGURA 7 - TESSERA PERSONALIZZATA PER LA MOSTRA

### 3.2 OBIETTIVI

Lo scopo che si prefigge il software è di rendere la visita la meno noiosa possibile. Com'è già stato menzionato, il pubblico cui si rivolge il programma è principalmente composto di bambini, siano essi parte di una scolaresca in visita al museo o di una famiglia che ha deciso di passare la domenica assieme ad antichi reperti Ming. L'obiettivo principe è, quindi, intrattenere in modo intelligente e costruttivo questi bambini mentre si aggirano per il museo. I giochi, inoltre, sono tematici e pensati per consolidare le nozioni apprese lungo la visita e magari arricchirle o completarle con contenuti di semplice assimilazione.

Il software doveva occuparsi della riproduzione di contenuti Flash con l'identificazione degli utenti giocanti e la connessione ad un database centrale dove salvare i punteggi.

Oltre a questo si rendeva necessario un secondo programma, diverso da quello con cui dialogava direttamente l'utente, per controllare il tutto e con il quale fosse possibile iniziare e terminare la sessione di gioco. Le tessere, infatti, dovevano essere consegnate all'inizio all'utente che, dopo aver terminato la mostra e magari giocato con le nostre postazioni, le riconsegnava ricevendo in cambio qualche gadget, magari in funzione dei punti ottenuti.



### 3.3 REQUISITI

I requisiti, che dovevano rispettare il sistema, possono essere riassunti con tre termini: integrazione, solidità e semplicità.

Il palazzo in cui si svolgeva la mostra era stato accuratamente addobbato per rimanere in tema; elettricisti, falegnami e carpentieri si erano dati da fare perché tutto fosse come richiesto in tempo per l'inaugurazione. L'impianto multimediale non poteva essere un'eccezione: doveva integrarsi senza creare elemento di contrasto eccessivo; così, per venire incontro a questa esigenza, abbiamo scelto di utilizzare dei monitor LCD appesi alle pareti e di nascondere i PC delle singole postazioni. Le stesse tessere RFID sono state camuffate per integrarsi all'ambiente attraverso degli adesivi che vi abbiamo applicato sopra mentre mobiletti di legno, realizzati per l'occasione, nascondevano i PC. All'utente finale perciò era visibile solo il monitor, un joystick per giocare e il lettore di schede.

Altro aspetto molto importante del sistema doveva essere la solidità e l'affidabilità. Se il tutto avesse smesso di rispondere o di funzionare, non solo la reputazione della mostra e questa stessa ne avrebbero risentito ma anche i clienti si sarebbero potuti lamentare. Per riuscire a raggiungere tale requisito si è cercato di chiudere il sistema nel modo migliore possibile con un numero di operazioni limitate e una minor varietà d'input. Ciò ha permesso una gestione più facile limitando i casi non previsti o da prevedere. È stata inoltre adottata una strategia per il controllo remoto del sistema e un metodo di segnalazione in caso di malfunzionamento. Il software, inoltre, teneva traccia di tutte le eccezioni che si sarebbero potute verificare e di tutti gli eventi di rilievo, in modo da poter conoscere la causa di un eventuale comportamento non desiderato.

Infine, bisogna tener conto della semplicità. Poiché la mostra era gestita da personale non tecnico e soprattutto era rivolta ai bambini, l'interazione con il software è stata ridotta al minimo al fine di semplificare e agevolare il più possibile. L'utilizzo delle tessere e del joystick si è rivelato perciò particolarmente funzionale sia per un discorso di affidabilità sia per la semplicità.



## 4 TECNOLOGIE E INFRASTRUTTURE DEL SISTEMA

### 4.1 MySQL

Il termine database, banca dati o base di dati, indica un insieme di archivi (un tipo di file) collegati secondo un particolare modello logico e in modo tale da consentire la gestione dei dati stessi da parte di particolari applicazioni software dedicate (DBMS – Database Management System –).

MySQL è un DBMS, sviluppato dall'Oracle Corporation con licenza open source, basato sul modello relazionale: i dati sono raccolti in tabelle collegate tra di loro da relazioni che creano dipendenze e correlazioni tra dati altrimenti indipendenti. Un modello relazionale fonda la sua struttura su due concetti chiave: l'entità e la relazione. Entrambi questi concetti si concretizzano in tabelle che raccolgono e aggregano informazioni in delle "celle" denominate *campi*.

MySQL ben s'integra con qualsiasi infrastruttura software e hardware. È un software libero che presenta una grande flessibilità con ottime prestazioni ed è disponibile anche un driver ufficiale che permette di accedere agevolmente alle funzioni per ambienti .NET.

Prima della realizzazione effettiva del database si crea un modello detto *entity-relationship* (ER) che permette di rappresentare concettualmente i dati; lo schema ER del database è rappresentato nella Figura 8 sottostante. Come si può vedere il DB è composto di due entità: Points e Rank; non collegate da alcuna relazione.

Points è una tabella temporanea che contiene i punti che gli utenti giocatori realizzano; la chiave della tabella è rappresentata dal campo tag che contiene l'UID letto della tessera dell'utente.

Rank, invece, è una tabella che contiene i punteggi realizzati dagli utenti e permette di risalire a quando questi siano stati realizzati; la chiave della tabella è un campo auto incrementante di nome idRank che aumenta a ogni nuova sessione di gioco. Nella tabella vengono anche memorizzati gli istanti d'inizio e fine gioco per la costruzione di eventuali classifiche.

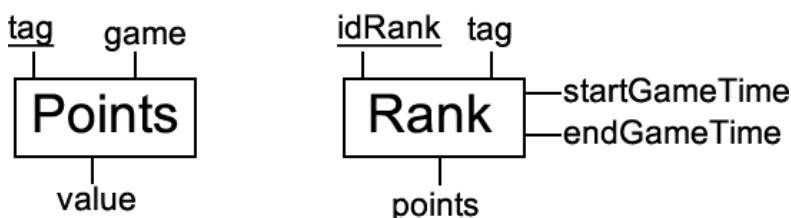


FIGURA 8 - ER DEL DB

Una volta creato lo schema ER si procede con la creazione delle tabelle.

Points		
CAMPO	TIPO	DESCRIZIONE
<b>tag</b>	varchar	Id del Tag dell'utente
<b>game</b>	int	Numero identificativo del gioco
<b>value</b>	varchar	Punti realizzati nel gioco

Rank		
CAMPO	TIPO	DESCRIZIONE
<b>idRank</b>	int	ID auto incrementante che serve da chiave per possibili usi futuri.
<b>tag</b>	varchar	Id del Tag con cui l'utente ha giocato
<b>points</b>	varchar	Punti realizzati
<b>startGameTime</b>	datetime	Data e ora di quando è iniziato il percorso di gioco.
<b>endGameTime</b>	datetime	Data e ora di quando è stato terminato il percorso di gioco.

## 4.2 C#

Il C# è un linguaggio di programmazione orientato agli oggetti sviluppato da Microsoft all'interno dell'iniziativa .NET. La sintassi del C# prende spunto da quella del Delphi, del C++, di Java e di Visual Basic e, soprattutto di questi ultimi due, prende i punti di forza. La semplicità nella stesura del codice e della realizzazione dell'interfaccia utente ne ha designato il successo. L'IDE (Integrated Development Environment) permette di realizzare ottime interfacce utente senza scrivere nemmeno una riga di codice; ciò è possibile grazie ai componenti standard, che vengono personalizzati tramite le proprietà specifiche dell'oggetto. Al programmatore non rimarrà che scrivere il codice delle azioni associate agli elementi dell'interfaccia senza curarsi del codice necessario per impostare l'aspetto voluto.

Un ulteriore punto di forza di questo linguaggio è rappresentato dalla possibilità di gestire agevolmente i giochi scritti in Flash, tramite un pratico componente realizzato dalla stessa Adobe Systems.

Si osservino ora alcuni termini chiave di questo linguaggio a oggetti che saranno usati in seguito; termini come: "metodo", "variabile", "Form" che ai non addetti ai lavori potrebbero essere oscuri.

Qualsiasi linguaggio di programmazione basa il suo funzionamento su "variabili" e "funzioni". Con variabile si fa riferimento a una porzione di memoria che è riservata per contenere il tipo di dato che si andrà a salvare. Essa è caratterizzata da un nome e spesso da un tipo che indica la natura del contenuto. La funzione, invece, è un costrutto che permette di raggruppare una sequenza d'istruzioni in un unico blocco, espletando così una determinata operazione sui dati del programma stesso.

Come sopra citato, C# discende anche da Java e, come tale, basa il suo funzionamento sulla programmazione a oggetti; quest'ultima è un paradigma di programmazione che permette di

definire oggetti software in grado di interagire con gli altri attraverso lo scambio di messaggi. La programmazione ad oggetti prevede di raggruppare in una zona circoscritta del codice sorgente (chiamata classe) la dichiarazione delle strutture dati e delle procedure che operano su di esse. Le classi, quindi, costituiscono dei modelli astratti, che a tempo di esecuzione sono istanziati per creare Oggetti software. Ecco un esempio chiarificatore: la *classe* è il progetto che descrive come costruire una casa e in che modi possiamo interagire con essa, mentre l'*oggetto* è la casa una volta che è stata costruita seguendo il progetto. Gli oggetti creati saranno dotati di attributi (variabili proprie dell'oggetto) e metodi (particolari funzioni) che permetteranno di interagire con esso, secondo quanto dichiarato nella rispettiva classe.

A un livello di astrazione maggiore appartiene il Form che è un termine usato per indicare l'interfaccia di un'applicazione che consente all'utente di inviare uno o più dati liberamente. I Form, che stanno alla base dell'aspetto del software, sono le finestre con cui l'utente interagisce in qualsiasi sistema operativo.

Dopo aver illustrato alcuni termini chiave del linguaggio, si può terminare con l'IDE. Visual Studio è un potente programma per lo sviluppo di applicazioni che supporta molti linguaggi prodotti dalla stessa Microsoft e non solo; grazie a esso creare applicazioni dal complesso aspetto visivo è semplice e rapido grazie agli elementi già pronti. Per utilizzare un componente di cui si ha bisogno ("bottone", "campo di testo", ecc.) è sufficiente trascinarlo nel Form per utilizzarlo.

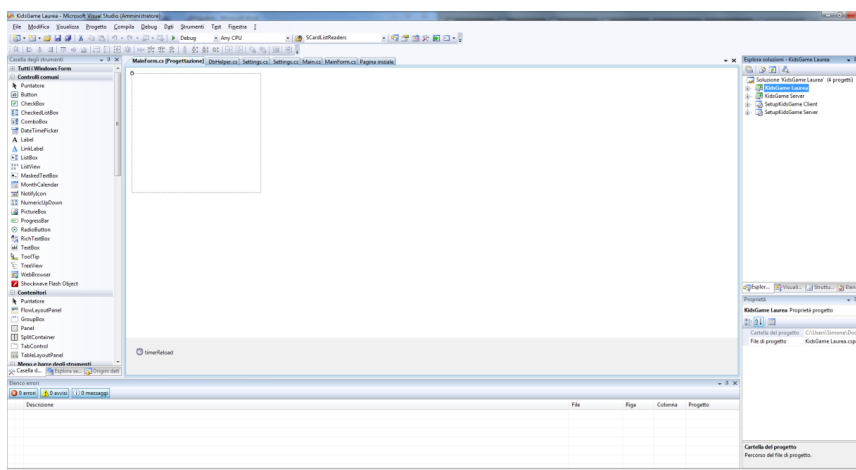


FIGURA 9 - VISUAL STUDIO

### 4.3 FLASH



Flash è un software per uso prevalentemente grafico che consente di creare animazioni vettoriali in modo semplice e rapido. È sviluppato dall'Adobe Systems (in precedenza Macromedia) ed è disponibile per le maggiori piattaforme software.

Flash permette di creare animazioni complesse e multimediali; all'interno dei fotogrammi che compongono l'applicazione, infatti, si possono inserire:

- Forme vettoriali, che sono gli oggetti principali con cui Flash permette di lavorare.
- Testo, sia statico sia dinamico, e caselle d'input per il testo
- Immagini
- Audio
- Video
- Altre animazioni Flash

Inoltre è dotato di un linguaggio di scripting interno molto potente denominato ActionScript. Questo linguaggio permette di applicare particolari effetti o animazioni agli oggetti e ai fotogrammi dell'animazione; grazie ad esso si possono creare infatti menu, sistemi di navigazione, GUI, siti web completi e giochi anche complessi.

Rappresenta la soluzione ideale per lo sviluppo di giochi e ne permette la realizzazione, con risultati spesso sorprendenti, in modo semplice e rapido.

### 4.4 LA RETE ADOTTATA

Per permettere ai PC di comunicare tra di loro, e in particolare per permettere alle postazioni client di comunicare con il server, si è scelto di adottare una semplice rete locale. Presso ogni postazione, infatti, erano già disponibili delle porte di rete e la connessione è avvenuta in modo semplice e rapido.

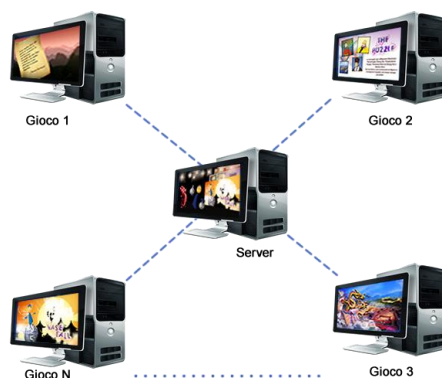


FIGURA 10 – ARCHITETTURA DELLA RETE.

## 4.5 LOGMEIN

LogMeIn è il nome di una società produttrice di un software per la connessione e la gestione dei PC in remoto; tra i prodotti offerti ce ne sono di varie tipologie che possono essere distinti per prezzi e opzioni che offrono: per il controllo remoto dei PC è sufficiente la versione gratuita.

Per la connessione è sufficiente registrarsi al sito internet e installare il software necessario, scaricabile dalla pagina del produttore, nella postazione da controllare. In seguito, collegandosi allo spazio apposito nel sito, dopo essersi identificati con username e password, si utilizzerà la voce apposita per stabilire la connessione e controllare il PC dal proprio browser. È uno strumento potente e versatile, indispensabile per controllare dei PC non fisicamente disponibili.

## 4.5 ELEMENTI DEL SISTEMA

Ogni postazione era composta di una serie di oggetti accuratamente scelti per adattarsi all'ambiente circostante.



Un monitor da 32 pollici da 1080p appeso alla parete per visualizzare i contenuti.



Un PC dell'ACER dalle piccole dimensioni in modo da poter essere nascosto dietro alle postazioni predisposte.



Un joystick per interagire con il sistema



Il lettore RFID per identificarsi e iniziare i giochi.





## 5 IL PROGRAMMA

Il software realizzato si compone di due sottodistribuzioni: una, lato client, per gestire i giochi montati sulle postazioni, e una, lato server, per gestire inizio e fine del gioco.

### 5.1 CLIENT

Il programma era stato costruito intorno alla necessità di poter riprodurre giochi in flash. Fortunatamente, Visual Studio fornisce un modo molto semplice per aggiungere questa funzionalità ai programmi. Con l'installazione nei PC di Flash Player, che abilita alla riproduzione di alcuni contenuti presenti nel Web, è stato installato nel sistema un componente utile per lo scopo. Per aggiungerlo al progetto è sufficiente cliccare con il tasto destro nella barra degli strumenti dell'IDE e scegliere la voce "Scegli elementi".

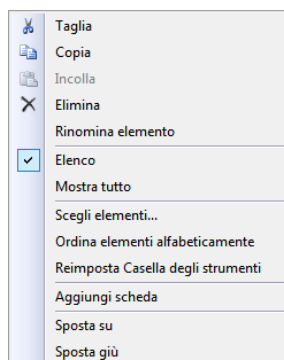


FIGURA 11 - MENU PER LA SCELTA DEI COMPONENTI DELLA BARRA DEGLI STRUMENTI

Nella schermata successiva, nella scheda dei Componenti COM, troviamo "Shockwave Flash Object" e, dopo averlo selezionato e cliccato sul tasto "OK", lo troviamo nella barra degli strumenti e non resta che trascinare il componente nel Form per utilizzarlo.

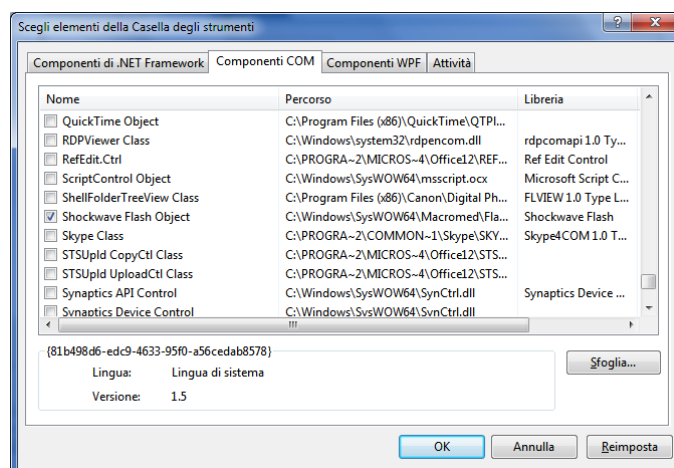


FIGURA 12 - FINESTRA PER LA SELEZIONE DEI COMPONENTI AGGIUNTIVI

Il componente *Shockwave Flash Object* è molto facile da usare: dispone di metodi e proprietà utili per controllare e gestire un file flash. Per riprodurre un contenuto di questo tipo è sufficiente impostare il valore della proprietà *Movie* con l'indirizzo sul disco del contenuto da

riprodurre. Sia “flashPlayer” il nome del componente appena aggiunto al Form e “urlFlash” la variabile che contiene l’indirizzo del gioco, il codice seguente ci permette di riprodurlo.

```
flash Player=urlFlash;
```

Questa serie di operazioni aggiunge al programma una funzionalità così potente e complessa che con altri ambienti avrebbe richiesto un maggiore dispendio di tempo ed energie. Per dargli la forma desiderata si devono modificare le proprietà che regolano l’aspetto e configurarlo in modo che vada a occupare tutto lo spazio disponibile del Form. Andando ad agire sulle proprietà di quest’ultimo riusciamo a rendere il programma privo dei classici pulsanti di chiusura e ridimensionamento della finestra e visualizzare a tutto schermo solo il Form, nascondendo quello che c’è sotto.

Il componente *Shockwave Flash Object* dispone di una funzione molto potente e utile: `FSCommand`; questa funzione utilizzabile nel codice C# viene richiamata dal codice del gioco Flash e permette la comunicazione tra i due sistemi altrimenti eterogenei. È appunto grazie a questa funzione che si riesce a conoscere il punteggio effettuato nel gioco Flash e a visualizzarlo successivamente.

Poiché il programma doveva riprodurre contenuti differenti per ogni postazione, è stata introdotta una schermata apposita per configurare la postazione; in essa è possibile impostare il numero di gioco, la possibilità di rigiocarlo, le schermate iniziali e finali e la directory del gioco. Andava, inoltre, indicato anche l’indirizzo di rete del server, dove è situato il database, e le credenziali per accedervi. Il *numero di gioco* identificava la postazione ed era indispensabile per distinguere e salvare i giochi nel database. La voce “rigiocabile” permetteva all’utente di riprovare e di migliorare o peggiorare la sessione precedente; nel DB, infatti, era salvata solo l’ultima partita; nel caso non ci fosse stata la spunta su questa voce, l’utente poteva giocare una sola volta e un ulteriore passaggio della tessera era ignorato.

In ogni postazione sono stati salvati localmente i contenuti come le immagini iniziali e finali da visualizzare e il gioco flash, oltre al software realizzato. Per accedere alla finestra, poiché l’applicazione occupava tutto lo schermo e non si potevano inserire bottoni, si è scelto di usare una combinazione di tasti; in particolare per accedere alla suddetta schermata bisognava premere “ctrl + y”.

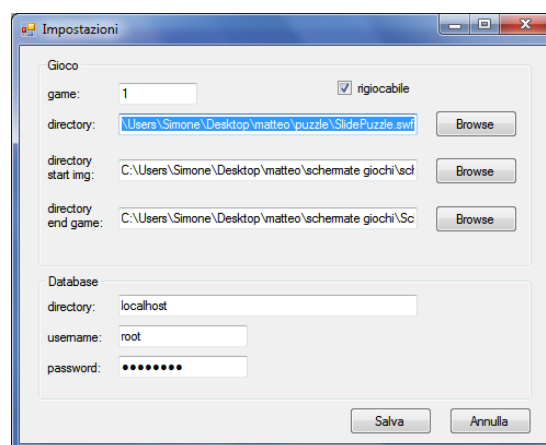


FIGURA 13 - SCHERMATA DELLE IMPOSTAZIONI

L'applicazione, occupando tutto lo schermo, nascondeva il sistema operativo sottostante e, una volta lanciata, visualizzava l'immagine scelta come iniziale.



FIGURA 14 - SCHERMATA INIZIALE DEL GIOCO

La schermata era visualizzata nell'attesa che qualche utente iniziasse il gioco passando la tessera sul lettore.



FIGURA 15 - PASSAGGIO DELLA TESSERA SUL LETTORE

Questo attivava l'evento che impostava il gioco e lo lanciava; permettendo all'utente di giocarci.



FIGURA 16 - SCHERMATA DEL GIOCO

Una volta terminato il gioco era visualizzato il punteggio e la schermata finale. Attraverso la funzione sopra menzionata, `FSCommand`, era ricevuto nel programma C# il punteggio effettuato e si procedeva nel comporre la schermata finale con il punteggio e salvando tale risultato nel Database.



FIGURA 17 - SCHERMATA FINALE DEL GIOCO

Dopo 10 secondi la schermata finale scompariva e veniva riproposta quella iniziale in attesa che un nuovo utente giocasse.

Al fine di rendere maggiormente affidabile il software e tempestive le operazioni di assistenza, è stato integrato un algoritmo in grado di salvare e inviare i problemi verificatisi. Tutte le eccezioni erano salvate in un apposito file di testo secondo uno schema fisso: data, descrizione del problema. In tal modo si poteva controllare l'uso che se ne era fatto e il tipo d'imprevisti in cui si poteva incorrere; ciò permetteva di capire quali miglioramenti implementare in usi futuri e dove prestare maggiore attenzione. In caso di arresti critici o problemi con il lettore era inviata una mail, in modo automatico, contenente in allegato il file log e una breve descrizione dell'evento che ne ha provocato l'invio.

Infine, il programma aveva una parte di codice che ne impediva più esecuzioni contemporanee nello stesso PC: lavorando con driver era rischioso permetterne un'eventuale esecuzione multipla.

## 5.2 SERVER

Il software del server era installato su di un PC presente all'ingresso; era stato nascosto in modo tale che l'utente vedesse solo lo schermo tv e un'operatrice si occupava della consegna e del passaggio della tessera sul lettore.

Un'immagine a tutto schermo, che introduceva l'utente ai giochi che sarebbero seguiti, era visualizzata nell'attesa che la tessera utente fosse inizializzata.



FIGURA 18 - SCHERMATA D' ATTESA PER INIZIARE UNA SESSIONE DEL GIOCO

Quando l'operatrice passava il Tag sul lettore il programma si occupava di cancellare i punteggi precedenti della tessera dal DB, eliminando la storia precedente, per permettere di giocare nelle postazioni successive del percorso.



FIGURA 19 - SCHERMATA D' INIZIO SESSIONE

Per gestire inizio e fine erano stati predisposti due lettori collegati allo stesso PC; questo per semplificare il lavoro dell'operatrice e ridurre al minimo i disagi dovuti all'inesperienza con il sistema. Il programma che girava sul server era più essenziale; mancava del componente per la riproduzione dei contenuti Flash e si limitava ad operazioni di lettura/scrittura del DB e delle tessere.

Un'apposita schermata permetteva di selezionare quale lettore iniziava e quale terminava la sessione e questa selezione veniva fatta da chi installava il sistema, normalmente, una volta sola; solo nel caso si fosse riavviato il sistema si sarebbe dovuto rifarlo.

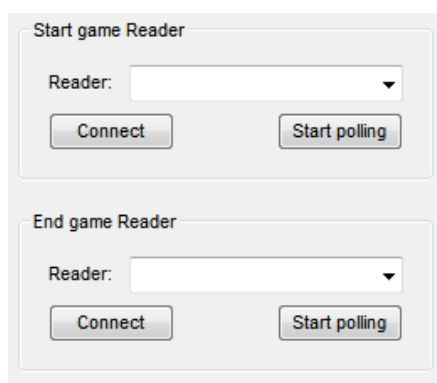


FIGURA 20 - SELEZIONE DEL LETTORE D' INIZIO E FINE SESSIONE

Anche su questo programma, sebbene fosse specifico e non andasse installato in altre postazioni, era stata predisposta un'apposita schermata di configurazione; le informazioni da inserire si limitavano a parametri per la connessione al DB come username, password e la directory del DB. A questa schermata come per il client si accedeva tramite la pressione di un'opportuna configurazione di tasti: "ctrl+y".

Nella schermata sottostante si può vedere come nel campo della directory sia indicato "localhost"; questa parola permette di indicare al sistema come destinazione il PC stesso. La possibilità di specificare la directory era stata introdotta in previsione di una possibile separazione tra la postazione server e il DB.

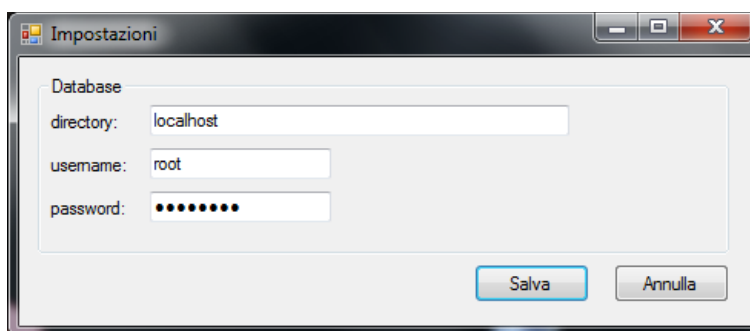


FIGURA 21 - SCHERMATA DELLE IMPOSTAZIONI DEL PROGRAMMA SERVER

Una volta terminata la sessione, l'utente riconsegnava la tessera e, dopo che l'operatrice l'aveva passata nell'apposito lettore, veniva visualizzata la schermata finale con i punteggi cumulativo realizzato.



FIGURA 22 - SCHERMATA FINE SESSIONE

Nel DB era salvato il punteggio complessivo di data d'inizio e fine per fini sia statistici sia promozionali; in un sito internet apposito venivano visualizzati i punteggi migliori realizzati nell'ultimo periodo.

### 5.3 INSTALLAZIONE DEL SISTEMA

L'installazione del software era costituita di alcuni componenti :

- Il file eseguibile che installava il software;
- I driver necessari al lettore;
- La cartella specifica della postazione contenente le immagini e il gioco flash da riprodurre.

Per l'installazione era stato creato un file eseguibile che si occupava di copiare nelle opportune directory i file oltre ad apportare le modifiche necessarie al sistema. Non era stato possibile integrare in tal eseguibile i file come le immagini, perché specifiche delle postazioni e nemmeno i driver, perché dotati di un file eseguibile proprio.





## 6 VALUTAZIONE E TESTING DEL SISTEMA

### 6.1 AFFIDABILITÀ E SOLIDITÀ DEL SISTEMA

Il software non è un oggetto statico e immutabile: una volta creato, raramente, permane invariato in un sistema, spesso si presta a miglioramenti e modifiche. Tuttavia, si devono definire dei parametri per chiarire quando un prodotto è pronto per essere rilasciato al cliente; non esiste una misura universale che quantifichi gli aspetti che un software deve avere per essere considerato di buona qualità. Esistono, però, degli aspetti da considerare che danno una stima, e che sono condivisi dai più, come l'affidabilità e la robustezza. Con il termine affidabilità di un software si fa riferimento a quanto raramente si manifestano dei malfunzionamenti; un programma affidabile sarà in grado di funzionare per molto tempo prima che si manifestino dei guasti. La robustezza "misura" in che modo un sistema si comporta con eventi imprevisi, non contemplate nelle specifiche.

Tale analisi è stata eseguita a sistema installato. Per più volte si è compiuto il percorso multimediale della mostra, spesso simulando comportamenti errati e volutamente assurdi. Il sistema, dopo alcune correzioni, è risultato affidabile e robusto secondo le aspettative e pronto per essere esposto al pubblico che ne avrebbe testato ancora più a fondo le capacità.

### 6.2 PROBLEMI RISCONTRATI

Nella realizzazione ci si era imbattuti in delle difficoltà di vario genere e spesso si è stati costretti ad adottare particolari accorgimenti per superarle.

Il progetto era nato a poca distanza dalla data d'inaugurazione prevista per la mostra; uno tra i primi problemi da affrontare era stato il tempo. Abbiamo dovuto scegliere velocemente che cosa utilizzare per soddisfare le richieste e metterci subito all'opera.

Dopo ore passate a visionare cataloghi online delle ditte produttrici di lettori RFID; si era optato per quelli che avevano un prezzo idoneo al budget previsto per il progetto. Una volta arrivati, però, abbiamo dovuto far fronte a un problema: capire come usarli. Come già detto, erano forniti di manuali ed esempi ma la connessione avveniva attraverso mezzi che potremmo definire "manuali". Per altri lettori di uguale tipologia esistono sistemi di connessione e controllo a più elevata astrazione, ad esempio attraverso OCX (OLE Control eXtension), che ne permettono l'utilizzo con una conoscenza limitata al minimo. La scarsa preparazione sull'argomento unita all'inesperienza della prima volta ci aveva obbligato a spendere del tempo prezioso per studiare manuali e documentazione di varia natura.

La difficoltà riscontrata nell'utilizzo non è stato l'unico inconveniente di questi lettori. In occasioni molto rare la lettura avveniva in modo errato e per qualche ragione il lettore

smetteva di funzionare. Tramite accorgimenti del software si erano limitati i danni e permesso al sistema di funzionare correttamente.

Altro punto difficile era stato prevedere tutti i possibili tipi di utilizzo non idonei del sistema. Nelle prime versioni, infatti, mescolando pressioni dei tasti del joystick con passaggi di tessera nel mentre, si generavano combinazioni imprevedibili che potevano portare all'instabilità e malfunzionamenti. Una buona parte del tempo era stata spesa a verificare che il software non presentasse di queste debolezze e a riparare quelle trovate.

Altra nota dolente era il sistema operativo stesso. Temevamo che potesse smettere di funzionare proprio a causa dell'utilizzo prolungato dei PC o generare eventi imprevedibili che potevano deteriorare il funzionamento del software creato. A titolo esemplificativo citiamo un possibile problema: il software era visualizzato a tutto schermo; se windows avesse visualizzato qualche tipo di messaggio questo si sarebbe potuto andare a sovrapporre alla nostra schermata deteriorando l'aspetto del prodotto o addirittura compromettendone l'utilizzo. Chiaramente per impedirlo sono state prese opportune contromisure nel software in grado di gestire ed evitare anche questi eventi e il sistema veniva riavviato ogni notte quando la mostra era chiusa.

## 7 CONCLUSIONI

Nella tesi è stata analizzata la tecnologia RFID e un progetto che ne sfrutta le potenzialità. L'RFID, che fino a pochi anni fa era considerata una tecnologia innovativa e rivoluzionaria, oggi viene ampiamente utilizzata nei più svariati ambiti ed è destinata ad una ulteriore crescita e sviluppo. Ampie possibilità si stanno aprendo nel mercato del pagamento elettronico e in particolare tramite cellulare; l'NFC (Near Field Communication), già presente in alcuni dispositivi, rappresenta una concreta evoluzione di questa tecnologia che presto entrerà ancor più nella nostra vita. L'obiettivo ultimo del progetto non si limitava alla realizzazione di un programma come attrazione per invogliare i clienti a entrare nella mostra di Treviso ma si collocava in un più ampio contesto; il software era parte attiva e elemento fondamentale della mostra. Infatti, l'esposizione non si era limitata a questo evento ma successivamente ha girato il mondo e il programma con essa. Questa tesi offre anche interessanti spunti di riflessione su come la tecnologia sia un fattore in grado di integrarsi in tutti gli ambienti, migliorandone gli aspetti e le potenzialità.



## TAVOLA DEGLI ACRONIMI

ACK	positive ACKnowledgement
ATS	Answer to select
DB	Database
DBMS	Database Management System
FRAM	Ferroelectric Ram
GUI	Graphical User Interface
HLTA	Comando di Halt per il PICC di tipo A
PCD	Proximity Card
PICC	Proximity Card
PPS	Protocol and Parameter Selection
RATS	Request for Answer To Select
UID	User Identifier
WUPA	Comando di Wake-Up per il PICC di tipo A



## APPENDICE

### Comunicazione con il lettore

L'interazione con il lettore si compone dei seguenti passi:

1. Definisco il contesto
2. Controllo che lettori sono connessi al PC
3. Scelgo il lettore con cui comunicare
4. Mi collego e registro il lettore nell'applicazione tramite degli identificativi
5. Leggo ciclicamente lo stato del lettore finché non risulta presente una tessera
6. Leggo la tessera e salvo l'ID del tag
7. Torno al punto 5 o termino il collegamento.

La funzione `SCardEstablishContext` stabilisce il contesto di gestione delle risorse (campo di applicazione) in cui vengono eseguite le operazioni di database. L'OS, infatti, ha un database di smart card per gestire le risorse e si deve comunicare al lettore le informazioni che gli permettano di dialogare con lo stesso. In questo caso specifico questo è fatto impostando il valore della variabile `dwScope` al valore 0 che indica che ogni operazione viene svolta nel dominio dell'utente. Principalmente l'attenzione va riposta nell'uso che verrà fatto della variabile `phContext`: è attraverso di essa che si è in grado di specificare il contesto al lettore.

```
public static extern int SCardEstablishContext (
    int dwScope,
    int pvReserved1,
    int pvReserved2,
    ref int phContext
);
```

Nel passo successivo si ottiene una lista di lettori connessi al sistema e a cui ci si può collegare; per farlo si usa la funzione `SCardListReaders`. Alla variabile `hContext` si deve impostare il valore, ottenuto precedentemente, della variabile `phContext`.

La Variabile `Readers` ci fornirà l'elenco dei lettori disponibili.

```
public static extern int SCardListReaders (
    int hContext,
    byte[] Groups,
    byte[] Readers,
    ref int pcchReaders
);
```

Passiamo ora alla connessione attraverso la funzione `SCardConnect` che si occupa di stabilire l'effettivo collegamento tra l'applicazione e il lettore. Questa funzione restituisce attraverso la variabile `phCard` un intero che identifica il tipo di tessera utilizzata con il lettore.

```

public static extern int SCardConnect (
    int hContext,
    string szReaderName,
    int dwShareMode,
    int dwPrefProtocol,
    ref int phCard,
    ref int ActiveProtocol
);

```

Per la lettura ci si affida alla funzione `SCardTransmit` che invia una richiesta di servizio alla smart card e aspetta di riceverne la risposta; se la lettura è andata a buon fine viene ritornato il valore zero altrimenti un altro valore intero ad indicare il tipo di errore verificatosi. La variabile `RecvBuff` ci fornisce un puntatore alle informazioni restituite dal lettore. `SendBuff` è una variabile il cui contenuto specifica al lettore il tipo di comando che deve eseguire; va configurata in modo coerente con le richieste APDU sopra descritte.

```

public static extern int SCardTransmit (
    int hCard,
    ref SCARD_IO_REQUEST pioSendRequest,
    ref byte SendBuff,
    int SendBuffLen,
    ref SCARD_IO_REQUEST pioRecvRequest,
    ref byte RecvBuff,
    ref int RecvBuffLen);

```

Una volta ricevuta la stringa si tratta di scomporla e analizzarla come discusso precedentemente nel capitolo 2.5 riguardante gli APDU. Dopo aver verificato il buon esito del comando si salva l'UID così ottenuto per tutte le operazioni future riguardanti l'utente e la sua identificazione.



## BIBLIOGRAFIA

[www.waazaa.org/14443/](http://www.waazaa.org/14443/)

[it.wikipedia.org/](http://it.wikipedia.org/)

[en.wikipedia.org/](http://en.wikipedia.org/)

[www.microsoft.com](http://www.microsoft.com)

[www.rfiditalia.it](http://www.rfiditalia.it)

[msdn.microsoft.com](http://msdn.microsoft.com)

[www.acs.com.hk](http://www.acs.com.hk)