



I.I.S. GUGLIELMO MARCONI – DALMINE (BG)

## BASK-TECH



*“L’informatica al servizio dello sport.”*

**Candidato:** *Ezequiel Damian Lopes*

**Classe:** *5<sup>^</sup> Bin*

**Indirizzo:** *Informatica*

**Anno Scolastico:** *2015/2016*

# INDICE

1. INTRODUZIONE PROGETTO .....	2
1.1 Come Funziona .....	3
2. SVILUPPO SOFTWARE .....	6
2.1 Analisi Requisiti .....	6
2.2 Sviluppo SW .....	7
2.2.1 Progettazione lato Client .....	7
2.2.2 Progettazione lato server .....	8
2.2.3 Progettazione DataBase .....	9
2.2.4 Progettazione sito PHP .....	11
2.2.5 Crittografia Password .....	13
3. ARCHITETTURA HARDWARE .....	14
3.1 Cosa è Arduino .....	14
3.2 Sensore E18-D80NK .....	15
4. PIANO DI PROGETTO .....	16
4.1 Piano di Gantt .....	17
4.1 PERT .....	18
5. CONCLUSIONI E SVILUPPI FUTURI .....	19

# 1. INTRODUZIONE PROGETTO

L'idea del progetto è nata dal desiderio di unire le cose che più mi piacciono fare: programmare al computer e giocare a pallacanestro.

L'obiettivo di questo progetto è semplicemente quello di dimostrare le mie competenze in ambito informatico e, soprattutto, quello di rendere più piacevole e simpatico il lavoro fatto a scuola.

Il progetto che ho realizzato è rappresentato da un canestro di basket tecnologico, in cui il giocatore deve cercare di fare centro con una pallina e, grazie al software sviluppato da me, il computer terrà conto dei canestri segnati; inoltre mediante una base di dati, si avrà la possibilità di vedere la propria posizione in una classifica contenente tutti i punteggi di tutti i giocatori! Un amministratore può eventualmente eliminare dei punteggi oppure direttamente un giocatore dalla base di dati.

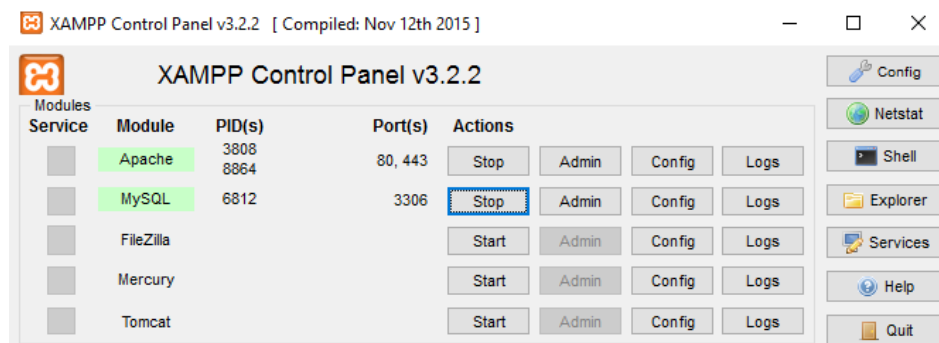
Ovviamente si potrebbe portare questo progetto in un canestro di dimensioni standard, implementandolo e migliorandolo ad esempio aggiungendo più sensori, in modo tale da sostituire la persona addetta al conteggio dei punti.



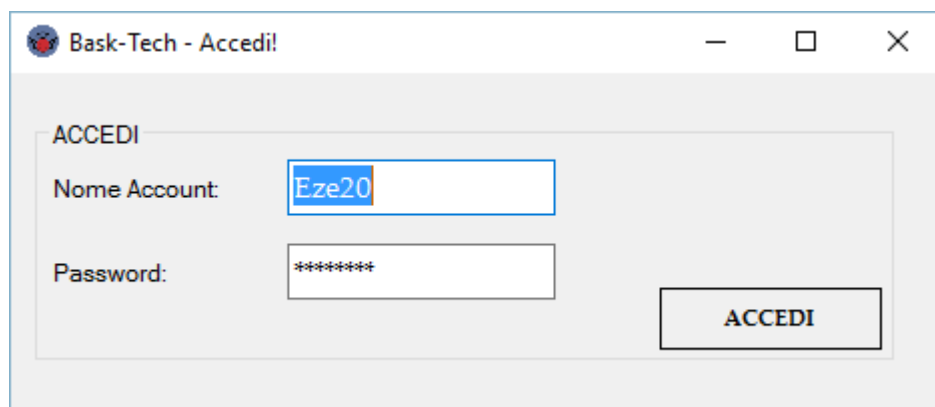
## 1.1 COME FUNZIONA

Andiamo ora ad analizzare come opera tale progetto, dal momento che avremo bisogno di diversi componenti software affinché esso possa eseguire tutte le istruzioni necessarie al suo funzionamento.

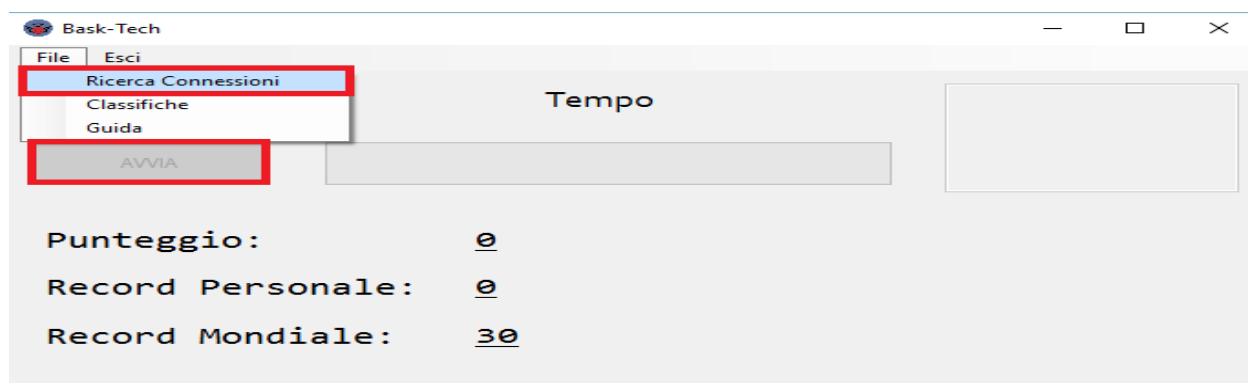
Innanzitutto facciamo partire tramite XAMPP i servizi Apache e MySQL, come in figura.



Il giocatore dovrà accedere con il proprio account, salvato in un DataBase MySQL, tramite una form di login in un'applicazione C#:



Nel caso in cui le credenziali siano corrette, si aprirà un'altra form che permetterà la connessione TCP/IP con Arduino (in ascolto sulla medesima porta) e, una volta instaurata questa, l'avvio del timer di gioco.



L'utente da adesso dovrà semplicemente tirare con una palla di piccole dimensioni cercando di fare canestro; in caso positivo il sensore E18-D80NK capterà il movimento, Arduino lo analizzerà e manderà tramite Socket il messaggio all'applicazione C# che, dopo averlo letto, incrementerà il punteggio del giocatore.

Una volta terminata la partita, il punteggio verrà automaticamente salvato sul DB MySQL; a questo punto l'utente potrà accedere sempre con il proprio account in una piattaforma sviluppata in PHP (Hypertext Preprocessor).

**ACCEDI!**

Username

Password

GO

Non hai ancora un account? Registrati!

**REGISTRATI!**

Username

Nome

Cognome

Data di nascita

June 2016						
Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Paese

Password

GO

Essa permetterà di visualizzare la classifica generale, la classifica per paese, la classifica per data della partita e i punteggi personali.

The screenshot shows the 'Visualizza' menu with options: 'I Tuo Punteggi', 'Punteggi per Paese', and 'Punteggi per Data Partita'. Below the menu, the user 'Luigi' is shown with the 'Classifica Mondiale Punteggi' table.

Posizione	Nome	Punti	Data Partita
1	Mario	30	02/06/2016 07:19:21
2	Luigi	22	02/06/2016 09:28:18
3	Mario	10	01/06/2016 10:21:30

The screenshot shows the 'Classifica Personale Punteggi' for user 'Mario'.

Posizione	Nome	Punti	Data Partita
1	Mario	36	30/05/2016 15:32:43
2	Mario	22	01/06/2016 08:13:19

The screenshot shows the 'Nome Paese' filter set to 'Argentina' and the 'Classifica Punteggi in Argentina' for user 'Mario'.

Posizione	Nome	Punti	Data Partita
1	Eze20	43	25/05/2016 21:15:23
2	Eze20	40	16/05/2016 18:30:00
3	Eze20	28	25/05/2016 22:48:41

The screenshot shows the 'Seleziona Data' calendar and the 'Classifica Punteggi del 2016-06-01' for user 'Mario'.

Posizione	Nome	Punti	Data Partita
1	Mario	22	01/06/2016 08:13:19



## 2. SVILUPPO SOFTWARE

### 2.1 ANALISI REQUISITI

Riguardo ai software utilizzati, ho usufruito delle potenzialità dei seguenti programmi:

- Visual Studio 2015, per sviluppare il lato Server dell'applicazione;
- Brackets, per sviluppare la piattaforma Web;
- Arduino, per sviluppare il lato Client dell'applicazione;
- XAMPP (X-platform, Apache HTTP Server, MariaDB, PHP, Perl), mediante il quale è possibile avere un'applicazione server capace di interpretare pagine web dinamiche in PHP.

The Apache http web server is an open source Web server application managed by the Apache Software Foundation. The server software is freely distributed, and the open source license means users can edit the underlying code to tweak performance and contribute to the future development of the program.

Apache will run on all major operating systems, it is most frequently used in combination with Linux.

Apache is the clear leader in the web server market, accounting for just under 42% of the total market, according to a February 2014 Netcraft survey.



## 2.2 SVILUPPO SOFTWARE

### 2.2.1 Progettazione lato Client

Il lato Client del progetto è rappresentato da Arduino; dopo aver montato la scheda Ethernet infatti esso può possedere un indirizzo IP che gli permetterà di comunicare con altri dispositivi all'interno, in questo caso, della stessa rete.

Il primo passo è quello di dichiarare le librerie necessarie all'utilizzo di questi protocolli;

```
1. #include <Ethernet.h>
2. #include <SPI.h>
```

La libreria <SPI.h> permette ad Arduino, usato come dispositivo principale, di comunicare con altri dispositivi SPI; Il Serial Peripheral Interface o SPI è un sistema di comunicazione tra un microcontrollore e altri circuiti integrati o tra più microcontrollori (nel nostro caso la Shield Ethernet).

Se si utilizza la scheda Ethernet di Arduino, la libreria <Ethernet.h> permette alla scheda Arduino di connettersi alla rete Internet.

Dopodiché Arduino resta in attesa finché gli viene assegnato un indirizzo IP e successivamente cercherà di connettersi al Server (dichiarato all'inizio del codice) che resterà in ascolto sulla porta 9000.

```
1. byte server[] = { 192, 168, 1, 4 };
2. // start the Ethernet connection:
3. if (Ethernet.begin(mac) == 0)
4. {
5.   Serial.println("Failed to configure Ethernet using DHCP");
6.   // no point in carrying on, so do nothing forevermore:
7.   for (;;)
8.     ;
9. }
10.
11. if (client.connect(server, 9000))
12. {
13.   Serial.println("connected");
14. }
15. else
16. {
17.   Serial.println("connection failed");
18. }
```

Una volta instaurata la connessione TCP/IP, Arduino invierà pacchetti contenenti il messaggio necessario all'incremento del punteggio del giocatore solamente quando quest'ultimo farà canestro.



## 2.2.2 Progettazione lato Server

Il lato Server del progetto è rappresentato da un'applicazione form sviluppata nel linguaggio C#; per prima cosa una form chiederà le credenziali che l'utente dovrà inserire, le quali nel caso in cui siano corrette, permetteranno l'avvio di un'altra form dedicata alla connessione con il Client e l'avvio della partita.

```
1. if (corretto)
2. {
3.     Form2 frm = new Form2(nome_account, punteggio_record_account, punteggio_record_tot);
4.     frm.Show();
5.     this.Enabled = false;
6. }
```

Per poter comunicare con il DB ho creato una classe DBConnect.cs contenente tutti i metodi necessari per la connessione e disconnessione con esso, per l'inserimento di dati e per la lettura di dati (OpenConnection, CloseConnection, Login...).

L'utente, a questo punto, dal MenuStrip selezionerà la voce 'Ricerca Connessioni' in modo da permettere al Client Arduino di potersi connettere con esso; a livello di codice di programmazione, un Thread 'ListeningThread' dichiarato globalmente verrà istanziato con la funzione 'void ListenForConnections()'. In questa funzione il TcpListener, ossia una classe che fornisce metodi semplici che attendono e accettano richieste di connessione in ingresso, viene avviato e rimane in attesa di connessioni. Una volta arrivata e accettata la richiesta tramite la funzione 'AcceptSocket()', il Socket 'ClientConnection' viene istanziato e sarà quello che ci permetterà di comunicare con Arduino.

L'oggetto 'DataStream' della classe NetworkStream, che fornisce metodi per l'invio e la ricezione di dati mediante Socket, farà riferimento al Socket 'ClientConnection'.

```
1. private void ricercaConnessioniToolStripMenuItem_Click(object sender, EventArgs e)
2. {
3.     ListeningThread = new Thread(new ThreadStart(ListenForConnections));
4.     ListeningThread.Start();
5. }
6. private void ListenForConnections()
7. {
8.     try
9.     {
10.         ConnectionListener = new TcpListener(IPAddress.Parse("192.168.1.4"), 9000);
11.         ConnectionListener.Start();
12.         ChangeTextBoxContent("Listening For Connections");
13.         ClientConnection = ConnectionListener.AcceptSocket();
14.         DataStream = new NetworkStream(ClientConnection);
15.         ChangeTextBoxContent("Connection Received");
16.     } catch (Exception ex) { MessageBox.Show(ex.Message); }
17. }
```

Una volta instaurata la connessione con il Client, il Server comunicherà con esso tramite Socket; il giocatore quindi premerà il bottone 'Avvia' che aprirà un Thread che terrà conto del tempo trascorso. A tempo scaduto (60 secondi) la partita finirà e il punteggio verrà automaticamente salvato in una base di dati e potrà essere visualizzato in un secondo momento.

```
1. DBConnect inserimentodb = new DBConnect();
2. string sqlFormattedDate = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
3. inserimentodb.InsertPunteggio(lbl_nomeaccount.Text, punteggio_corrente, sqlFormattedDate);
4. MessageBox.Show("Punteggio salvato!");
```

## 2.2.3 Progettazione DataBase

A DataBase is an organised collection of data and used when you need to store lots of data. Data is organised into fields (column) and records (rows).

The big benefit is that you can search data quickly. There are two types of DataBase:

- Relational DB: it stores data in separate tables and fields. These are predominant today because they offer tremendous flexibility and power.
- Non-Relational DB: data is organised into one table, which can be viewed by opening one data file.

La tipologia che ho scelto è il RDBMS (Relational DataBase Management System), che si basa appunto sul modello relazionale e sull'algebra relazionale (= linguaggio procedurale, ossia una descrizione della procedura da attuare per ottenere un risultato).

Il DataBase che ho sviluppato è molto semplice, composto da tre tabelle collegate tra di loro tramite relazioni, secondo il seguente modello logico:

- **Giocatore** (Username, Nome, Cognome, DataNascita, Paese)
- **Login** (FK Username, Psw, Ruolo)
- **Punteggio** (FK Username, DataPartita, Punti, ID)

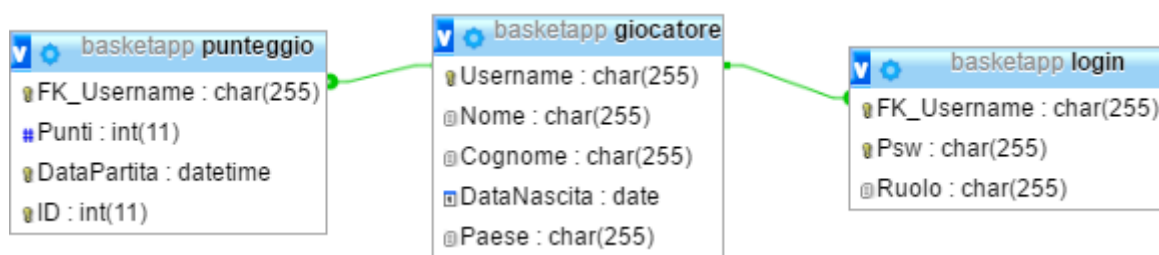
Le parole in grassetto identificano le tabelle, ossia le entità del DataBase, mentre le parole sottolineate rappresentano le chiavi primarie di ogni entità, che devono essere univoche all'interno della stessa tabella; nel caso dell'entità Punteggio, avendo una coppia di chiave

primarie, è possibile che un campo della stessa colonna si ripeta ma non è possibile che la coppia si ripeta all'interno di questa tabella.

Brevemente, un Giocatore ha uno Username che deve essere univoco e inoltre possiede dei campi anagrafici che servono per riconoscerlo e per essere attribuito a specifiche classifiche (Classifiche per Paese). Ogni giocatore possiede dunque delle credenziali per il login, rappresentate dallo stesso Username e da una Password (il ruolo serve semplicemente per distinguerlo dall'amministratore).

Infine ogni giocatore può effettuare diverse partite, dunque avere uno o più punteggi che potranno essere individuati tramite lo Username del giocatore e la data (nel formato YYYY/mm/dd HH:mm:ss) in cui è stata giocata la partita.

Dal punto di vista concettuale, il DataBase può essere rappresentato seguendo il Modello Entity-Relationship (anche detto modello entità-relazione oppure modello ER), ossia un modello per la rappresentazione concettuale dei dati ad un alto livello di astrazione, come in figura:



Tutte le tabelle del DataBase che ho sviluppato sono in Terza Forma Normale, quindi anche in Prima e Seconda Forma Normale:

- 1FN, perché gli attributi hanno un dominio semplice (forma atomica) e non ci sono record duplicati;
- 2FN, perché tutti gli attributi non chiave dipendono funzionalmente e completamente dall'intera chiave;
- 3FN, perché tutti gli attributi non chiave dipendono direttamente dall'intera chiave.

Il DataBase rispetta vari vincoli di integrità dei dati, ossia delle proprietà che devono essere soddisfatte dalle istanze di una base di dati.

Un vincolo è rappresentato dalla chiave primaria, ossia un campo che serve a identificare univocamente ogni record di una tabella. Due righe distinte non possono avere lo stesso valore sui campi scelti come primary key.

Una chiave univoca (unique key) è simile alla chiave primaria, con la particolarità che gli attributi coinvolti possono assumere il valore nullo (vedi ID nella tabella Punteggio).

Il vincolo interrelazionale (o extrarelazionali) più utilizzato è il vincolo di integrità referenziale: gli attributi di una data tabella possono assumere soltanto dei valori specificati in un'altra tabella. Vi sono appositi costrutti per questo vincolo (ad esempio la Foreign Key, utilizzate nelle tabelle Login e Punteggio).

## 2.2.4 Progettazione Sito PHP

Il sito per funzionare necessita che i servizi Apache e MySQL siano attivi sul computer e che esso sia salvato sul disco C all'indirizzo *C:/xampp/htdocs*; questo perché la macchina deve saper interpretare il linguaggio PHP e perché il sito deve comunicare e reperire le informazioni dal DataBase.

Il sito si compone principalmente da due parti:

- Login/Registrazione
- Home Page

Il Login è composto da una form realizzata tramite CSS3, in cui l'utente deve inserire le proprie credenziali per poter accedere alla propria Home Page; se una persona non possiede ancora un'utente, potrà registrarsi tramite un'altra form realizzata, in questo caso, tramite CSS3 e Bootstrap (quest'ultimo per il calendario, in quanto bisogna salvare la data di nascita dell'utente). Nel caso in cui le credenziali siano corrette, l'utente sarà reindirizzato nella Home Page che avrà caratteristiche diverse in base al Ruolo dell'utente che vuole accedere.

Se l'utente è un 'Giocatore', la Home Page, realizzata con Bootstrap, sarà composta da una toolbar dropdown che permetterà di navigare all'interno del sito:

- Nella 'Home' sarà possibile visualizzare la classifica generale di tutti i punteggi;
- Nell'opzione 'I tuoi Punteggi', come suggerisce il titolo, sarà possibile visualizzare i migliori punteggi dell'utente loggato;
- Nell'opzione 'Punteggi per Paese', sarà possibile scegliere tramite una combobox il Paese di cui si vuole vedere la classifica;
- Nell'opzione 'Punteggi per Data Partita', sarà possibile scegliere tramite un calendario un giorno qualsiasi e visualizzare la classifica di quel determinato giorno;

Se l'utente è un 'Admin', la Home Page, sarà composta da due form:

- La prima permetterà all'amministratore di eliminare dal DataBase uno o più punteggi dei giocatori;
- La seconda permetterà all'amministratore di eliminare dal DataBase uno o più giocatori con i relativi punteggi.

L'intero sito è caratterizzato dalle sessioni; dal momento che il protocollo http è statless (ossia non è in grado di tenere traccia di particolari dati trasmessi tra client e server), in PHP è stato implementato da un meccanismo di sessioni, dove per 'sessioni' si intendono tutte le interazioni avvenute tra browser e server fino alla chiusura del browser stesso. In questo modo si potrà tenere memorizzato l'accesso dell'utente anche se, per diversi motivi, verrà chiusa la scheda del browser.

Questo meccanismo funziona grazie alle variabili di sessione, ossia delle variabili globali a tutte le pagine, che servono per memorizzare i dati essenziali alle sue attività. Nel mio caso, le variabili globali sono \$\_SESSION['Username'] e \$\_SESSION['Ruolo'].

A livello di codice, queste variabili sono implementate nelle diverse pagine nella seguente maniera:

- Session.php:

```
1. $user_session=$_SESSION['Username'];
2. $rule_session=$_SESSION['Ruolo'];
3. if(!isset($user_session))
4. {
5.     $connection->close(); // Closing Connection
6.     header('Location: Accedi.php'); // Redirecting To Home Page
7. }
```

- Accedi.php (Se la variabile non è vuota, allora esiste un utente già loggato e quindi viene reindirizzato verso la pagina Benvenuto.php):

```
1. <?php
2. session_start();
3. if(isset($_SESSION['Username'])){
4. header("location: Benvenuto.php");
5. }
6. ?>
```

- Benvenuto.php (Includendo Session.php si esegue il codice della medesima pagina, quindi viene controllato se esiste un utente loggato, altrimenti si viene reindirizzati verso la pagina di Login):

```
1. <?php
2. include('Session.php');
3. ?>
```

## 2.2.5 Crittografia Password

Un nuovo utente, dovendosi registrare, deve inserire dei dati strettamente personali come, ad esempio, la Password; essa deve essere salvata in maniera sicura, in modo da evitare che persone o software di terze parti possano appropriarsene senza il consenso del diretto interessato.

Per provvedere a ciò, il linguaggio PHP fornisce moltissime funzioni di criptaggio che rendono incomprensibili e illeggibili i dati sensibili; una di queste è la funzione di *hash*, ossia un algoritmo matematico che trasforma dei dati di lunghezza arbitraria (messaggio) in una stringa binaria di dimensione fissa chiamata valore di hash o impronta del messaggio.

Gli algoritmi usati a questo proposito sono unidirezionali (one-way), quindi difficili da invertire, questo permette alle funzioni crittografiche di hash di trovare ampio utilizzo negli ambiti di sicurezza informatica come: nelle firme digitali, autenticazione dei messaggi oppure, come nel nostro caso, per la crittografia delle credenziali private degli utenti nelle applicazioni web.

La funzione di hash è resistente alle collisioni, ossia se si modifica anche solo un bit del messaggio, l'output della funzione cambia a valanga.

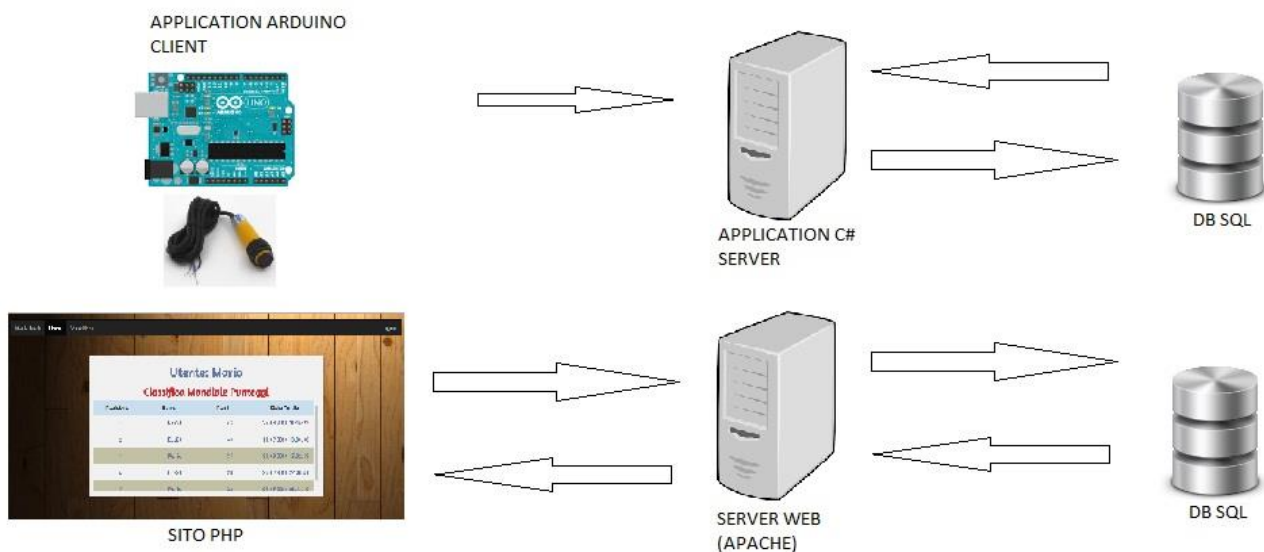
Qui sotto riporto la linea di codice utilizzata per far fronte a questo problema:

```
1. $password= hash ("sha256", $_POST['Password']);
```

Con il termine SHA (Secure Hash Algorithm) si indica una famiglia di cinque diverse funzioni crittografiche di hash; come ogni algoritmo di hash, l'SHA produce un "impronta del messaggio" con lunghezza pari al numero indicato di fianco (nel nostro caso, SHA-256 produce un valore di hash pari a 256 bit).

Utilizzando queste funzioni, rispettiamo lo standard ISO 27001 riguardante la sicurezza e, nello specifico, andiamo a garantire il primo e il secondo dei tre punti principali che possiamo abbreviare con la sigla RID (Riservatezza, Integrità e Disponibilità); più precisamente garantiamo che l'informazione sia disponibile solo a coloro che ne hanno l'autorizzazione e che sia quella generata dal sistema, senza nessuna modifica.

### 3. ARCHITETTURA HARDWARE



Il client Arduino, dopo essersi connesso con l'applicazione Server, semplicemente manda dei messaggi a quest'ultimo senza ricevere nessuna risposta, in quanto sarebbe superfluo.

Il server deve connettersi e scambiare messaggi con il DataBase SQL più volte durante l'esecuzione dell'applicazione, ad esempio per il Login e per il salvataggio dei punteggi.

Per quanto riguarda la piattaforma Web, si ha bisogno di un Server Web (in questo caso di Apache) per poter interpretare il linguaggio PHP e quindi permettere la comunicazione con il DataBase per ottenere le informazioni necessarie.

#### 3.1 Cosa è Arduino e la Shield Ethernet

Arduino è una piattaforma hardware low-cost programmabile, con cui è possibile creare circuiti "quasi" di ogni tipo per molte applicazioni.

Quando lo colleghiamo ad una fonte di alimentazione (ad esempio la USB del PC o anche una comunissima Batteria da 9V) si accende e avvia il programma caricato dall'IDE a loop infinito.

Questo continua fino a quando non lo si uccide a meno che non togliamo la batteria o stacciamo il cavo.

I Pin sono connettori che hanno molteplici utilizzi; i più importanti sono:

- PIN uscita corrente a 3.3V
- PIN uscita corrente a 5V



- Analog in - PIN input analogici, restituiscono un valore da 0 a 1023.
- Digital - PIN digitali programmabili per essere input o output, percepiscono se è presente o no corrente restituendo LOW se non c'è corrente e HIGH se c'è corrente.

La scheda Arduino ha un pulsante di Reset. Premendolo si collegherà temporaneamente il pin di reset a massa e quindi sarà possibile riavviare l'esecuzione di qualsiasi codice caricato in Arduino.

Inoltre, ci sono queste schede chiamate Shield; esse sono schede pre-costruite che si adattano sopra alla scheda Arduino e forniscono ulteriori funzionalità.

La shield che ho utilizzato è l'Ethernet Shield: essa permette di collegare Arduino alla "rete", per poter recuperare e caricare dati e risorse da/a Internet o da una semplice rete locale.

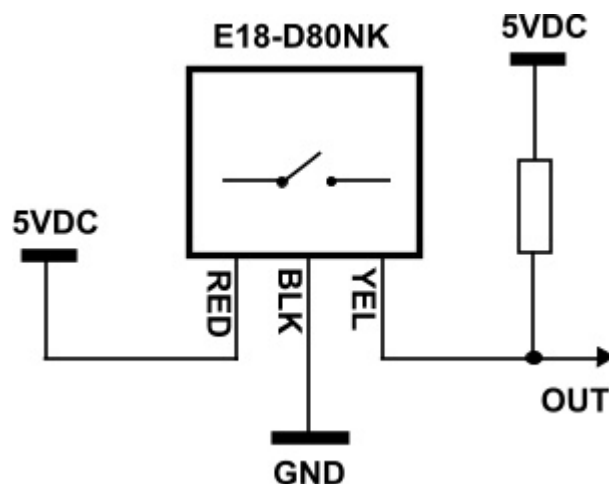
Il nome è W150, che corrisponde a quello del controllore presente sulla scheda.

L'installazione, dal punto di vista hardware è davvero semplice, in quanto occorre posizionarla sopra Arduino Uno, inserendo i PIN in modo corretto. Sarà possibile comunque usare i PIN per i nostri progetti, dal momento che esiste un contatto tra la scheda Ethernet e Arduino Uno.

### 3.2 Sensore E18-D80NK

Il modulo è costituito da un emettitore e da un sensore a infrarossi. Quando un ostacolo viene rilevato, l'uscita digitale viene attivata. Un trimmer permette di settare la distanza di intervento. E' presente inoltre un led di notifica.

Connessioni: rosso 5V, nero GND, giallo uscita digitale.



## 4. PIANO DI PROGETTO

Come ogni prodotto, anche un prodotto software possiede un proprio ciclo di vita (life cycle), ovvero una serie di fasi che ne descrivono l'evoluzione dalla sua nascita fino al suo abbandono. Affinché la produzione del software possa essere considerata di qualità è necessario seguire le indicazioni contenute nelle normative internazionali rilasciate dall'ISO (International Organization for Standardization).

Il ciclo di vita di un prodotto software viene formalizzato nel documento ISO/IEC 12207:2008 "System and software engineering – Software life cycle processes", il quale prevede tre gruppi: processi primari, processi di supporto, processi organizzativi.

I processi primari sono di certo quelli più significativi in quanto descrivono come produrre il software, mentre gli altri due gruppi servono di supporto.

La vita dei processi primari può essere schematizzato su una linea temporale (timeline).

Quelli utilizzati nel mio progetto sono i seguenti:

- Avvio (Analisi), che rappresenta appunto l'analisi dei requisiti e dei materiali che mi sarebbero serviti per completare il progetto;
- Pianificazione del Progetto e Reperimento Materiali;
- Realizzazione, che si suddivide nella progettazione di ogni singola parte del progetto finale (DataBase, Client, Server, Sito, Montaggio Hardware);
- Monitoraggio e Manutenzione, ossia controllare che il software funzioni correttamente e, nel caso contrario, correggere eventuali bug e/o migliorare le prestazioni generali;
- Chiusura, nel mio caso preparare i documenti in cui specifico ciò che ho fatto durante il periodo di costruzione del progetto.

## 4.1 Piano di Gantt

Nome	Data d'inizio	Data di fine
♀ ◉ Avvio	01/03/16	02/03/16
◉ Analisi Requisiti	01/03/16	02/03/16
♀ ◉ Pianificazione	03/03/16	13/04/16
◉ Reperimento Materiali	03/03/16	13/04/16
◉ Pianificazione SW	03/03/16	04/03/16
♀ ◉ Realizzazione	07/03/16	15/04/16
◉ Progettazione DB	07/03/16	08/03/16
◉ Progettazione Applicazione C# (Server)	07/03/16	01/04/16
◉ Progettazione Applicazione Arduino (Client)	07/03/16	01/04/16
◉ Progettazione Sito PHP	07/03/16	01/04/16
◉ Montaggio Scheda Arduino e Verifica Funzi...	14/04/16	14/04/16
◉ Montaggio Struttura Canestro	14/04/16	14/04/16
◉ Montaggio Totale	15/04/16	15/04/16
♀ ◉ Monitoraggio/Manutenzione	18/04/16	16/05/16
◉ Test e Correzioni	18/04/16	16/05/16
♀ ◉ Chiusura	17/05/16	30/05/16
◉ Preparazione Documenti	17/05/16	30/05/16

Lo strumento utilizzato per costruire questa tabella è il software Gantt Project.

Inizialmente ho dovuto pensare a tutto ciò che avrei necessitato per svolgere tutte le attività destinate al compimento del progetto.

Dopodiché ho dovuto attendere l'arrivo dei materiali acquistati su Amazon (Arduino, Shield Ethernet di Arduino, Sensore E18-D80NK, Canestro da Camera), che sono arrivati in date diverse e contemporaneamente ho iniziato a pianificare i tempi di sviluppo del progetto.

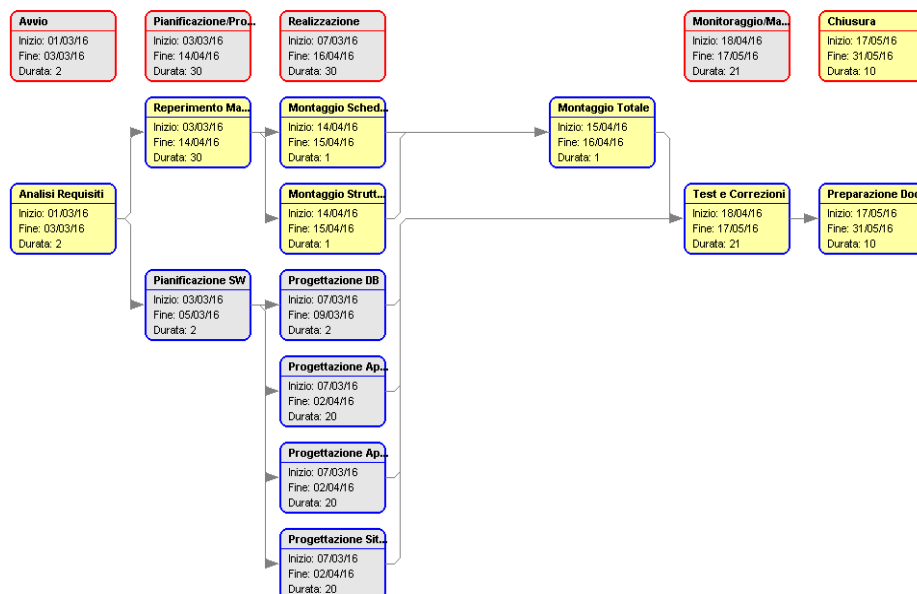
Mentre aspettavo i materiali, ho iniziato a progettare il DataBase e a sviluppare l'applicazione Server, l'applicazione Client e il sito PHP.

Una volta arrivati i componenti ho potuto finalmente montare il tutto e, nell'attività successiva, testare il funzionamento del progetto; ovviamente come tutte le prime prove e verifiche vi sono stati dei piccoli problemi che però ho risolto prontamente.

Infine ho preparato la documentazione per spiegare il funzionamento di tutto il progetto.

## 4.2 PERT

Il PERT (detta anche stima a tre valori) è un metodo statistico di determinazione dei tempi delle attività di progetto (ma può essere applicato anche ai costi). Rispetto alla semplice stima a valore singolo, il metodo presuppone la determinazione di valori di stima ottimale, probabile e pessimistico che risultano più adeguati a valutare tempi e costi di attività di progetto che presentano incertezza o complessità.



Il tempo totale impiegato per sviluppare il progetto è stato di quattro mesi; bisogna tenere conto però del fatto che in una spedizione ho avuto un ritardo di circa tre settimane dal momento che il fornitore da cui avevo comprato la Shield Ethernet di Arduino si è ritirato da Amazon causando quindi un ritardo all'intero progetto. La fase del 'Reperimento Materiali' infatti è un Punto Critico del mio Piano di Progetto.

## 5. Conclusioni e Sviluppi Futuri

Per rendere più funzionale il mio progetto si può pensare di:

- Collocare un tabellone sopra il canestro per rendere maggiormente visibile il punteggio in real-time;
- Modificare l'applicazione in modo da creare difficoltà diverse (raggiungere dei Checkpoint per poter andare avanti a giocare ecc.);
- Impostare un secondo sensore infrarossi per maggiore accuratezza nel rilevamento dei movimenti;
- Sostituire la Ethernet Shield con una Wi-Fi Shield per evitare l'utilizzo di cavi e quindi per maggiore comodità;
- Possibilità di personalizzare la piattaforma Web per ogni utente (sfondo, preferenze...);
- Impostare una rete DMZ con una macchina Server contenente un DataBase per memorizzare i dati globalmente e non localmente;

Dal punto di vista economico, si può pensare di utilizzare Bask-Tech all'interno di sale giochi, ovviamente riportando il tutto su una struttura di dimensioni maggiori, con il costo di circa 50 centesimi a partita con la durata di un minuto.

Per concludere, ritengo che l'informatica possa essere utilizzata in tutti gli ambiti della vita e farne buon uso può essere un vantaggio per tutti.