Beat The Piano

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.2.1 Spiegazione elementi tabella dei requisiti: 7

2.3 Use case 8

2.4 Pianificazione 9

2.5 Analisi dei mezzi 9

2.5.1 Software 9

2.5.2 Hardware 9

3 Progettazione 9

3.1 Design dell’architettura del sistema 10

3.2 Design dei dati e database 10

3.3 Design delle interfacce 11

3.4 Design procedurale 13

4 Implementazione 14

5 Test 14

5.1 Protocollo di test 14

5.2 Risultati test 15

5.3 Mancanze/limitazioni conosciute 15

6 Consuntivo 15

7 Conclusioni 15

7.1 Sviluppi futuri 15

7.2 Considerazioni personali 15

8 Glossario 15

9 Bibliografia 16

9.1 Bibliografia per articoli di riviste: 16

9.2 Bibliografia per libri 16

9.3 Sitografia 16

10 Allegati 16

# Introduzione

## Informazioni sul progetto

**• Allievo:** Ivan Ostroumov

**• Docente:** Geo Petrini

**• Scuola e classe:** SAMT Sezione informatica, I3BD

**• Materia:** Modulo 306

**• Data inizio consegna:** 05.09.2025 – 29.12.2025

## Abstract

## Scopo

Il progetto ha l’obiettivo principale di sviluppare un videogioco musicale educativo, interattivo, divertente e coinvolgente. Attraverso l’uso di un’interfaccia digitale, il gioco permette agli utenti di esercitarsi nel sviluppare il senso del ritmo e migliorare la coordinazione tra mani e occhi, simulando l’esperienza reale di suonare uno strumento.

Dal punto di vista didattico, serve a me per imparare a usare il game engine Unity, espandere le mie conoscenze sul design grafico e esercitarsi col linguaggio di programmazione C#.

Tra l’altro mi aiuterà a imparare a gestire un progetto usando le diverse tecniche fornite come il waterfall, i diversi diagrammi, eccetera.

# Analisi

## Analisi del dominio

**BEAT THE PIANO** è un software interattivo che unisce il concetto dei video MIDI al gameplay dei rhythm game. L’utente deve premere i tasti corretti sulla tastiera del computer nel momento in cui le note, che cadono dall’alto, raggiungono i tasti del pianoforte virtuale.

Il programma è pensato per tutti gli utenti che vogliono imparare il ritmo o semplicemente divertirsi, senza dover possedere una tastiera MIDI o conoscenze musicali. Ogni nota del pianoforte viene associata a un tasto della tastiera del PC (esempio: C# = 2, D = W), rendendo il gioco intuitivo e accessibile.

Attualmente esistono software simili come **Synthesia** o **Piano From Above**, ma richiedono l’uso di una tastiera MIDI e non prevedono una mappatura diretta con la tastiera del computer. **BEAT THE PIANO** si distingue proprio per questo: offre un’esperienza completamente giocabile solo con la tastiera del PC, combinando apprendimento musicale e divertimento.

## Analisi e specifica dei requisiti

Il progettista, dopo aver ricevuto il mandato, in collaborazione con il committente redige una lista di requisiti. Durante questi incontri, tramite interviste (da inserire nei diari), il progettista deve cercare di rispondere alle seguenti domande:

* Quali sono i bisogni del committente?
* Quali funzioni deve svolgere il prodotto?
* Come devono essere implementate?
* L’utente, come vorrebbe/dovrebbe interagire con il prodotto?
* Come verrà utilizzato il prodotto?
* Che tipo di interfaccia si immagina?
* Che prestazioni minime deve fornire il prodotto?
* Che grado di sicurezza deve avere il prodotto?
* …

In base alla lista dei requisiti e all’analisi degli stessi, il progettista redige una *specifica dei requisiti* in cui elenca e descrive in modo dettagliato quali sono le funzionalità che il prodotto fornirà. La specifica dovrebbe essere abbastanza dettagliata da poter essere utilizzata come base per lo sviluppo, ma non troppo; ad esempio non dovrebbe contenere dettagli di implementazione, o definizioni dettagliate dell’interfaccia grafica a meno che questi non siano considerati cruciali. Non si deve scordare che i requisiti non rappresentano delle attività bensì delle caratteristiche che il prodotto dovrà possedere.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-01 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Interfaccia Grafica | | | | |
| **Note** | L’utente deve poter visualizzare tutte le funzionalità e pagine dell’applicativo. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di un bottone per andare nella pagina di gioco. | | | | |
| **002** | Si necessita di un bottone per andare nella pagina di scelta di livello. | | | | |
| **003** | Si necessita di un bottone per andare nella pagina delle impostazioni. | | | | |
| **004** | Si necessita di un bottone per andare nella pagina dei credits. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-02 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Interfaccia Grafica Impostazioni | | | | |
| **Note** | L’utente deve poter visualizzare tutte le impostazioni che può cambiare. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di un bottone per andare nella pagina principale. | | | | |
| **002** | Si necessita di un volume slider per impostare il volume. | | | | |
| **003** | Si necessita di un color picker per impostare il colore delle note. | | | | |
| **004** | Si necessita di un bottone per cancellare i salvataggi. | | | | |
| **005** | Si necessita di un’immagine di pianoforte del gioco con ogni tasto cliccabile, dopo il click si assegna il tasto rispettivo sulla tastiera. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-03 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Interfaccia Grafica Credits | | | | |
| **Note** | L’utente deve poter visualizzare chi ha attribuito alla creazione del gioco. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di poter vedere chi ha creato le diverse parti del gioco. | | | | |
| **002** | Si necessita di un bottone per andare nella pagina principale. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-04 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Interfaccia Grafica Scelta livello | | | | |
| **Note** | L’utente deve poter visualizzare i diversi livelli che può giocare. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di un bottone per andare nella pagina principale. | | | | |
| **002** | Si necessita di un bottone per ogni livello presente per giocarlo. | | | | |
| **003** | Si necessita di uno sprite di lucchetto per ogni livello non ancora sbloccato. | | | | |
| **004** | Si necessita di una scritta con il nome per ogni livello. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-05 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Interfaccia Grafica Gioco | | | | |
| **Note** | L’utente deve poter visualizzare il processo di gioco. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di un background. | | | | |
| **002** | Si necessita di uno sprite di pianoforte con su ogni tasto il tasto associato da schiacciare. | | | | |
| **003** | Si necessita dei blocchi cadenti per far capire quando e quale nota suonare. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-06 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Interfaccia Grafica Risultati | | | | |
| **Note** | L’utente deve poter visualizzare il suo risultato di gioco | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di tre stelle che simboleggiano il risultato. | | | | |
| **002** | Si necessita di un bottone per tornare nella pagina principale. | | | | |
| **003** | Si necessita di un bottone per riprovare il livello. | | | | |
| **004** | Si necessita di un bottone per andare al prossimo livello. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-07 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Classe Nota | | | | |
| **Note** | Ci deve essere la classe che componerà le diverse melodie. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di una caratteristica spawnTime. | | | | |
| **002** | Si necessita di una caratteristica duration. | | | | |
| **003** | Si necessita di una caratteristica nota. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-08 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Classe Music | | | | |
| **Note** | Ci deve essere la classe su qui sono basati i livelli. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di una caratteristica name. | | | | |
| **002** | Si necessita di una caratteristica difficulty. | | | | |
| **003** | Si necessita di una caratteristica notes | | | | |
| **004** | Si necessita di una caratteristica tempo | | | | |
| **005** | Si necessita di un metodo calcolaPunteggio che confronta due Music. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-09 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Script Interpretator | | | | |
| **Note** | Ci deve essere lo script che ha la funzione Play creando i blocchi cadenti | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di un algoritmo che crea i blocchi leggendo le caratteristiche di Music. | | | | |
| **002** | Si necessita di un algoritmo che cancella i blocchi leggendo le caratteristiche di Music. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-10 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Script playerController | | | | |
| **Note** | Ci deve essere uno script che gestisce i controlli dell’utente | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di una keymap che salva le relazioni tra tasti di pianoforte e tastiera. | | | | |
| **002** | Si necessita di una funzione per cambiare i tasti associati. | | | | |
| **003** | Si necessita di una funzione per suonare la nota schiacciata. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-11 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Livelli | | | | |
| **Note** | Ci devono essere i livelli (tutti per solo 2 octave) | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di trovare due midi facili. | | | | |
| **002** | Si necessita di trovare due midi medi. | | | | |
| **003** | Si necessita di trovare due midi difficili. | | | | |
| **004** | Si necessita di trovare un midi difficilissimo. | | | | |
| **005** | Si necessita convertirli tutti nelle classi del programma. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito** | Req-12 | **Priorità** | 1 | **Versione** | 1.0 |
| **Nome** | Salvataggi | | | | |
| **Note** | Deve essere possibile salvare il progresso dell’ utente. | | | | |
| **Sotto requisiti** | | | | | |
| **001** | Si necessita di uno script per salvare tutti i dati. | | | | |
| **002** | Si necessita di uno script per caricare tutti i dati salvati. | | | | |
| **003** | Si necessita di uno script per aggiungere la cifratura in mezzo a salvare e caricare. | | | | |

### Spiegazione elementi tabella dei requisiti:

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

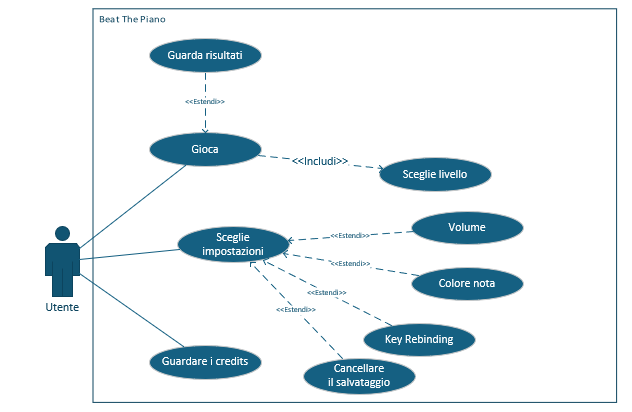
**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case



L’utente può giocare ai vari livelli dopo avergli scelti,

## Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

|  |
| --- |
| Figura 1: Esempio di diagramma di Gantt. |

Se si usano altri metodi di pianificazione (p.es. scrum), dovranno apparire in questo capitolo.

## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

Unity, C#, Photopea, Mockflow, Visual Studio 2022.

### Hardware

Computer con Windows 11, mouse, tastiera, monitor, SSD.

Caratteristiche pc:

Processore 13th Gen Intel(R) Core(TM) i7-13700 2.10 GHz

RAM installata 32.0 GB 4800MHz

Tipo sistema Sistema operativo Windows 11 a 64 bit, processore basato su x64

GPU NVIDIA T400 4GB

# Progettazione

Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## Design dell’architettura del sistema

Descrive:

* La struttura del programma/sistema lo schema di rete...
* Gli oggetti/moduli/componenti che lo compongono.
* I flussi di informazione in ingresso ed in uscita e le relative elaborazioni. Può utilizzare *diagrammi di flusso dei dati* (DFD).
* Eventuale sitemap

## Design dei dati e database

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

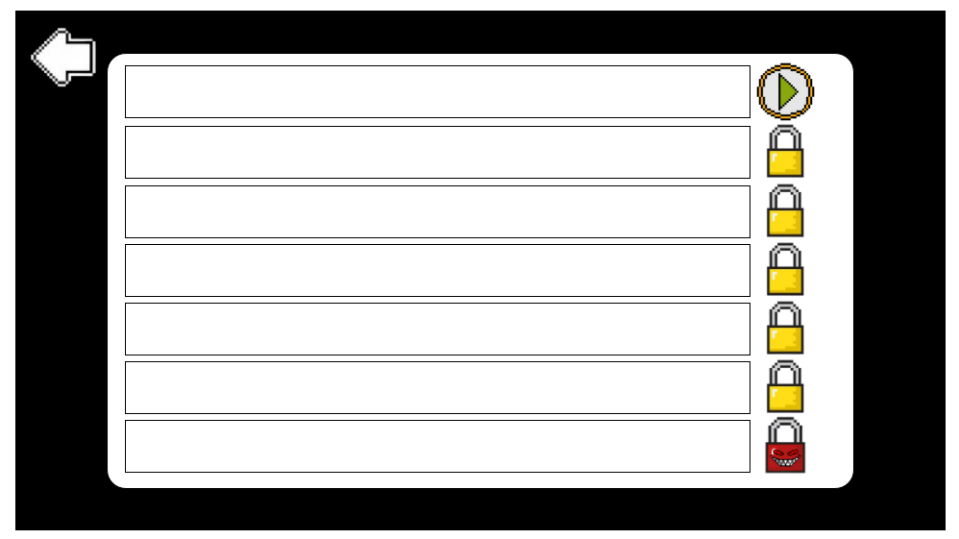
Se il diagramma E-R viene modificato, sulla doc dovrà apparire l’ultima versione, mentre le vecchie saranno sui diari.

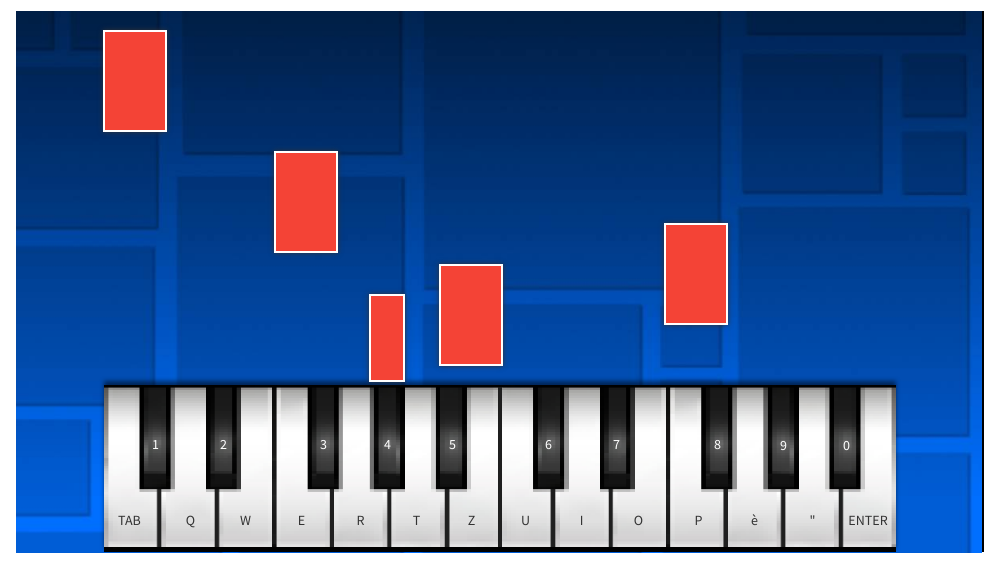
## Design delle interfacce













## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente - Print Screen - di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre, dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Glossario

Inserite una semplice tabella con due colonne che spieghi i termini specifici del progetto (lista dei termini in ordine alfabetico A-Z)

Esempio:

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| AJAX | **Asynchronous JavaScript And XML**: una tecnica che permette di eseguire richieste ed ottenere dati da una pagina web in modo asincrono. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo.

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o QdC
* Prodotto
* …