

# POLITECNICO DI TORINO

Dipartimento di Ingegneria Strutturale, Edile e Geotecnica

**Corso di Laurea Magistrale  
in Ingegneria Edile**

Tesi di Laurea Magistrale

## **Sviluppi del Modello BIM: Aumento di LOD, la modellazione MEP e l'utilizzo della Realtà Aumentata per la manutenzione**



**Relatore**

*Prof. Anna Osello*

**Candidato**

*Raffaele Basile*

Luglio 2018

# Sommario

<b>INDICE DELLE FIGURE</b> .....	4
<b>ABSTRACT</b> .....	7
<b>1 INTRODUZIONE</b> .....	9
<b>1.1 Caso studio: La Torre Regione Piemonte</b> .....	10
<b>1.1.1 Il Centro Servizi</b> .....	11
<b>1.2 Obiettivi da raggiungere</b> .....	12
<b>2 METODOLOGIA</b> .....	14
<b>3 IL BIM</b> .....	19
<b>3.1 Cos'è il BIM?</b> .....	19
<b>3.2 Le curve di MacLeamy</b> .....	20
<b>4 AUMENTO DI LOD</b> .....	22
<b>4.1 Cos'è il LOD</b> .....	22
<b>4.2 Applicazione al caso pratico</b> .....	26
<b>4.2.1 Impostazione delle famiglie</b> .....	27
<b>4.2.2 Creazione dei giunti</b> .....	28
<b>4.2.3 Modellazione in Advance Steel</b> .....	30
<b>4.2.4 Abaco delle connessioni strutturali in Revit</b> .....	33
<b>4.2.5 Abaco delle connessioni strutturali in Advance Steel</b> .....	34
<b>4.2.6 Confronto della modellazione degli elementi strutturali con la normativa</b> 36	
<b>5 MODELLAZIONE MEP</b> .....	38
<b>5.1 Organizzazione dei file</b> .....	38
<b>5.2 La modellazione</b> .....	40
<b>5.2.1 L'inserimento delle famiglie</b> .....	40
<b>5.2.2 La creazione dei circuiti elettrici</b> .....	40
<b>5.2.3 Gli abachi dei circuiti elettrici</b> .....	44
<b>5.2.4 Confronto della modellazione delle luci con la normativa</b> .....	45
<b>6 LA REALTA' AUMENTATA</b> .....	48
<b>6.1 Cos'è la Realtà Aumentata</b> .....	48
<b>6.1.1 Storia della Realtà Aumentata</b> .....	49
<b>6.1.2 Tipi di Realtà Aumentata</b> .....	49
<b>6.2 Applicazione al caso studio</b> .....	53
<b>6.2.1 Esportazione dei modelli Revit in fbx</b> .....	53
<b>6.2.2 Test 1: Utilizzo della Realtà Aumentata con Marker</b> .....	55
<b>6.2.3 Test 2: Utilizzo della Realtà Aumentata con Marker per ambiente</b> .....	61
<b>7 IMPORTAZIONE DELLE DATE AGGIORNATE IN REVIT</b> .....	93
<b>7.1 La creazione dei parametri condivisi</b> .....	93

7.2	Esportazione del database dal sito Altervista.....	95
7.3	Impostazione del DB Link in Revit .....	96
7.4	Unione dei database e creazione dell'abaco delle manutenzioni.....	98
8	<b>CONCLUSIONI E SVILUPPI FUTURI.....</b>	<b>101</b>
8.1	Conclusioni e sviluppi futuri sul processo di interoperabilità e datasharing 101	
8.2	Conclusioni e sviluppi futuri del modello strutturale .....	102
8.3	Conclusioni e sviluppi futuri del modello MEP .....	102
8.4	Conclusioni e sviluppi futuri dell'applicazione per la manutenzione in Realtà Aumentata .....	103
9	<b>RINGRAZIAMENTI .....</b>	<b>105</b>
10	<b>BIBLIOGRAFIA .....</b>	<b>106</b>
11	<b>SITOGRAFIA.....</b>	<b>107</b>
12	<b>ALLEGATI.....</b>	<b>109</b>
12.1	Rapporto giunto.....	109
12.2	Abaco delle connessioni strutturali.....	115
12.3	Distinta dei bulloni .....	116
12.4	Abaco dei circuiti elettrici.....	117
12.5	Consumi Applicazione.....	120
12.6	UnityARHitTestExample.cs.....	122
12.7	Clock.cs.....	123
12.9	<b>SCRIPT PHP .....</b>	<b>126</b>
12.9.1	db.php .....	126
12.9.2	load.php .....	127
12.9.3	save.php .....	127

## INDICE DELLE FIGURE

Figura 1: Grattacielo della Regione Piemonte.....	10
Figura 2: Centro Servizi .....	11
Figura 3: Workflow .....	14
Figura 4: Hardware utilizzati.....	16
Figura 5: Software utilizzati .....	17
Figura 6: Workflow della progettazione BIM – Fonte: <a href="http://www.sperastudio.it/?page_id=2658">http://www.sperastudio.it/?page_id=2658</a> .....	19
Figura 7: Curve di MacLeamy – Fonte: <a href="http://www.bis-lab.eu">http://www.bis-lab.eu</a> .....	20
Figura 8: Estratto dalla Norma LOD Spec 2017 Part I - LOD 300 .....	23
Figura 9: Estratto dalla Norma LOD Spec 2017 Part I - LOD 350 .....	23
Figura 10: Estratto dalla Norma LOD Spec 2017 Part I - LOD 400 .....	23
Figura 11: LOD per elementi strutturali (UNI 11337-4:2017).....	25
Figura 12: Posizione all'interno del workflow - Aumento di LOD .....	26
Figura 13: Scala antincendio - LOD C (LOG 300) .....	26
Figura 14: Percorso famiglie tedesche certificate.....	27
Figura 15: Elenco connessioni importate all'interno del modello strutturale .....	28
Figura 16: Centro Giunti – Esempio di “Splice Joint” di un controvento.....	29
Figura 17: Giunti Scale Antincendio .....	29
Figura 18: Risultato Giunti (Valori Fittizi) .....	30
Figura 19: Esportazione da Revit in Advance Steel .....	30
Figura 20: Centro Giunti in Advance Steel .....	31
Figura 21: Dettaglio dei giunti dei controventi della scala antincendio – LOG E (LOD 400) .....	31
Figura 22: Dettaglio delle fondazioni. A sinistra fondazioni della scala antincendio a destra dell'asilo nido – LOG E (LOD 400) .....	32
Figura 23: Dettaglio dei giunti delle travi poste in copertura – LOG E (LOD 400) .....	32
Figura 24: Stralcio dell'abaco delle connessioni strutturali in Revit .....	33
Figura 25: Numerazione Elementi.....	34
Figura 26: Estratto dalla distinta dei bulloni. Allegato 12.3.....	35
Figura 27: Schema organizzativo dei file .....	38
Figura 28: Visualizzazione 3D del modello con individuazione dei workset mediante diversi colori .....	39
Figura 29: Visualizzazione grafica degli elementi del modello assegnati ai relativi workset .....	39
Figura 30: Workset utilizzati per l'impianto elettrico.....	39
Figura 31: Proprietà di un tipo di luce a controsoffitto .....	40
Figura 32: Estratto del file "PR-3-CEP-CSSZ1.dwg" – Elenco dei circuiti collegati al quadro elettrico .....	41
Figura 33: Creazione Circuito - Esempio di un aula Conferenza in ambiente di modellazione BIM (Revit).....	42
Figura 34: Schema logico circuito.....	43
Figura 35: Gerarchia dei componenti .....	43
Figura 36: Estratto dell'abaco dei circuiti elettrici.....	44
Figura 37: Descrizione LOD dei corpi illuminanti (UNI 11337-4:2017) .....	45
Figura 38: Posizione all'interno del workflow - Esportazione del modello e creazione dell'applicazione .....	48

Figura 39: Esempio di un edificio visualizzato in AR con il metodo dei marker – Fonte: <a href="http://www.augmentend.com">www.augmentend.com</a> .....	50
Figura 41:Marker tridimensionale – Fonte : Video Youtube di Parth Anand - Augmented Reality Shopping. ....	51
Figura 41: Apple Watch visualizzato in AR tramite marker tridimensionale .....	51
Figura 42: Esempio di marker per ambiente – Fonte: <a href="http://www.apple.com/arkit/">www.apple.com/arkit/</a> .....	52
Figura 43: Esempio di marker ibrido – Fonte: <a href="http://techtrends.tech/tech-trends/gatwick-beacons/">http://techtrends.tech/tech-trends/gatwick-beacons/</a> .....	52
Figura 44: Posizione all'interno del workflow - Esportazione .....	53
Figura 45: Importazione AR Camera .....	55
Figura 46:Impostazioni Player Settings in Unity3D .....	55
Figura 47: Vuforia Configurator .....	56
Figura 48:Licenza sul sito di Vuforia.....	57
Figura 49: Creazione Image Target.....	57
Figura 50: Add Target .....	58
Figura 51: Punti noti dell'Image Target.....	58
Figura 52: Inspector dell'Image Target.....	59
Figura 53: Risultato Allineamento Image Target e modello strutturale.....	60
Figura 54 QR Code test Vuforia.....	60
Figura 55: Logo Apple ARKit.....	61
Figura 56:Importazione Apple ARKit.....	62
Figura 57: Mesh Compression del file architettonico .....	63
Figura 58: Inspector dei materiali.....	64
Figura 59: Hierarchy dell'ARKit .....	64
Figura 60: Modelli importati nella scena.....	65
Figura 61:Unity AR Hit Test Example script.....	66
<i>Figura 62: Logo Xcode .....</i>	<i>66</i>
<i>Figura 63: Impostazioni in Unity3D prima dell'esportazione in Xcode .....</i>	<i>68</i>
<i>Figura 64: Schermata di Xcode .....</i>	<i>69</i>
<i>Figura 65:Statistiche di Debug .....</i>	<i>69</i>
<i>Figura 66:Applicazione installata sull'iPhone7.....</i>	<i>70</i>
Figura 67: QR Code del modello in scala 1:10 .....	70
<i>Figura 68: Utilizzo CPU.....</i>	<i>70</i>
Figura 69: Script per la visualizzazione dei modelli .....	71
Figura 70: Pulsanti creati.....	71
Figura 71:Collegamento dei pulsanti alle funzioni desiderate .....	72
Figura 72:Unity AR Hit Test Example script modificato.....	73
Figura 73: Focus Square Script .....	74
Figura 75: Script collegati al tasto "Posiziona Modello" .....	75
Figura 75: Enable Object Script .....	75
Figura 76: Risultati ottenuti.....	75
Figura 77: Date della manutenzione programmata reperiti su internet .....	76
Figura 78: Menu dedicato per ogni elemento.....	77
Figura 79: LookAtCamera.cs .....	77
Figura 80:Stralcio dello script Clock.cs .....	78
Figura 81: Screenshot di un menu riguardante le luci.....	78
Figura 82: File salvati all'interno dell'iPhone utilizzato come test dell'applicazione .....	80
Figura 83: SQLite Logo.....	81
Figura 84: Maintenance Table.....	82
Figura 85: SELECT * FROM Maintenance.....	83

Figura 86: Contenuto del database .....	83
Figura 87: Script per la gestione del database con SQLite.....	84
Figura 88: Struttura del database .....	86
Figura 89: objManager.cs .....	87
Figura 90:Stralcio dello script MenuManager.cs .....	88
Figura 91: Gerarchia delle scene .....	89
Figura 92: Script per cambiare scena .....	90
Figura 93: Script per tornare al Main Menu .....	91
Figura 94: Funzione assegnata ai tasti.....	91
Figura 95: Posizione all'interno del workflow - Importazione dei dati all'interno del modello BIM .....	93
Figura 96: Elenco dei parametri condivisi relativi all'illuminazione.....	94
Figura 97: Tabella per la creazione dei parametri di progetto.....	94
Figura 98: Elementi attualmente presenti nel database .....	95
Figura 99:Impostazioni per l'esportazione del database in formato CSV .....	96
Figura 100: Database di Revit visualizzato in Access.....	97
Figura 101: Dati importati nel database collegato con Revit .....	98
Figura 102: Importazione del database aggiornato.....	98
Figura 103: Esito dell'importazione.....	99
Figura 104: Legenda degli output.....	99
Figura 105: Visualizzazione delle proprietà istanza dalla vista 3D.....	100
Figura 106: Utilizzo RAM.....	120
Figura 107: Utilizzo Batteria .....	120
<i>Figura 108: Frames per Second.....</i>	121
<i>Figura 109: Utilizzo CPU .....</i>	121
Figura 110: UnityARHitTestExample.cs .....	122
Figura 111: Script Clock.cs .....	123
Figura 112: Script Menumanager.cs.....	125

## ABSTRACT

L'industria delle costruzioni sta attraversando un periodo di trasformazione legato all'innovazione tecnologica. L'obiettivo oggi è quello di recepire dalla realtà sempre più dati per poi elaborarli ed utilizzarli per altri scopi. Nel campo dell'edilizia un ruolo fondamentale lo ricopre la metodologia di progettazione BIM grazie alla quale è possibile gestire tutti i dati e le informazioni relative ad un progetto.

Il presente elaborato tratta lo studio dei possibili sviluppi del modello BIM. Il caso studio scelto è il Centro Servizi antistante il grattacielo della Regione Piemonte. Partendo dal modello sviluppato durante il tirocinio è stato sviluppato il modello strutturale aumentando il livello di dettaglio; si è implementato il modello elettrico ed è stata sviluppata un'applicazione per la manutenzione sfruttando la realtà aumentata.

Particolare attenzione è stata posta a quest'ultimo aspetto, in quanto, ci si è approcciati ad un tipo di tecnologia nuova e in costante sviluppo.

Infine è stato studiato un metodo per far ritornare i dati, acquisiti in loco tramite l'applicazione, all'interno del modello BIM di partenza nella maniera più automatica possibile.

The construction industry is going through a changing period thanks to technological innovation. Nowadays the aim is to transpose more and more reality data in order to process and use them for other purposes. The design method “ BIM - Building Information Modeling” plays a key role in the construction sector. Because of it, it is possible to manage all the data and information related to a project.

This paper is about the study of the possible developments of the BIM model. The chosen case is the Service Center located on the opposite side of the skyscraper of the Piedmont Region. A structural model has been developed on the model improved during the internship, increasing the degree of detail. The electric model has been implemented and an app for the maintenance has been developed by using the augmented reality. Furthermore, special attention was paid on this last aspect because there was an approach to an in-progress technology. Finally, a method in order to bring acquired data back as automatically as possible has been analysed



# 1 INTRODUZIONE

Il presente elaborato di tesi fa parte di un più ampio lavoro comprendente anche il tirocinio curriculare svolto durante la prima parte dell'anno accademico 2017/2018.

Infatti il punto di partenza di questo lavoro è stato proprio il modello prodotto durante il primo semestre dell'anno accademico.

Si è scelto di lavorare sul medesimo caso studio per poter dedicare più tempo a questo progetto di quanto se ne sarebbe dedicato se si fosse lavorato solo nel tirocinio o solo nella tesi. Nelle ore di tirocinio è stato solo modellato il Centro Servizi antistante al grattacielo della Regione Piemonte; durante il lavoro di tesi, invece, partendo dal modello, è stato aumentato il LOD (Level of Detail) della parte strutturale, è stato modellato il sistema elettrico e infine è stata creata un'applicazione per tablet e smartphone per fare indagini non invasive in loco sfruttando le potenzialità della Realtà Aumentata (AR).

Si è scelto di utilizzare la Realtà Aumentata invece della Realtà Virtuale, perché, pensando ai costi e agli strumenti necessari per fare ciò, è risultato molto più economico e accessibile un'applicazione per dispositivi mobili, piuttosto che utilizzare visori per la realtà virtuale, i quali, sono molto costosi e necessitano a loro volta di computer con elevate prestazioni per poter reggere una tale mole di informazioni.

Il presente elaborato sarà suddiviso nelle seguenti parti:

- Aumento del LOG del C al LOG E (dal LOD 300 al LOD 400 secondo la normativa americana) del modello strutturale;
- Modellazione MEP (Mechanical, Electrical and Plumbing) dell'impianto elettrico;
- Sviluppo di un'applicazione per dispositivi mobili per ispezioni in loco non invasive e per la registrazione delle date nelle quali vengono eseguite le manutenzioni.
- Importazione dei dati salvati all'interno del modello Revit di partenza per la creazione di un abaco delle manutenzioni.

## 1.1 Caso studio: La Torre Regione Piemonte

“Fulcro dell’intero complesso, la torre per gli uffici è stata concepita per essere un edificio di riferimento per la comunità regionale e per tutti i cittadini, in grado di dialogare con gli edifici circostanti di importanza storica per la città, il Lingotto e l’Oval e il futuro Parco della Salute, della Ricerca e dell’Innovazione di Torino. Con duecentocinque metri e quarantuno piani fuori terra, la torre è realizzata in cemento armato e vetro, ed è caratterizzata dalle più innovative tecniche di risparmio energetico e di sostenibilità ambientale.



*Figura 1: Grattaciolo della Regione Piemonte*

Al suo interno verranno collocati tutti gli uffici della Regione attualmente distribuiti nelle 27 sedi del territorio cittadino, per un totale di oltre **2000** dipendenti. Sulla sommità della torre, negli ultimi due piani del grattaciolo, è prevista la realizzazione di una “Casa del Piemonte”. Uno spazio verde, con un giardino a cielo aperto, alberi da frutto, possibilità di ristorazione e di osservazione del panorama, aperto a piemontesi e turisti con cadenza regolare. Un grande corte accoglierà la base della torre ai due livelli interrati. Al piano terra la hall d’ingresso, con accesso al palazzo da via Nizza, ospiterà la reception e lo spazio espositivo della Regione.”<sup>[1]</sup>

Oltre al grattaciolo, il progetto prevede anche una corte interrata, dei parcheggi interrati, sviluppati su 3 piani, e un centro servizi, oggetto del presente lavoro di tesi, antistante alla torre e collegata ad essa mediante una passerella in acciaio e vetro.

---

<sup>1</sup> Fonte: [www.regione.piemonte.it/sedeunica/edifici.htm](http://www.regione.piemonte.it/sedeunica/edifici.htm)

### 1.1.1 Il Centro Servizi



Figura 2: Centro Servizi

Il centro servizi si sviluppa su 5 piani fuori terra e presenta al piano terra una hall e un asilo nido.

L'asilo nido si sviluppa per solo un piano in altezza mentre la hall presenta uno sviluppo di 2 piani fuori terra.

Al 2° piano vi sono 2 sale conferenza e un auditorium al quale si può accedere anche dal 3° piano. A questo piano vi sono 2 ul-

teriori sale conferenze e la sala regia dell'auditorium. Al 4° ed ultimo piano vi sono i locali tecnici, una biblioteca e una caffetteria.

Sia al 2° che al 3° piano vi è la possibilità di accedere al grattacielo mediante una passerella che collega i due corpi di fabbrica.

L'involucro dell'edificio è caratterizzato da componenti opachi per i primi due piani fuori terra e da una facciata vetrata per i restanti piani.

Anteposti alla facciata vi sono delle lamiera modulari che aiutano a schermare la luce solare. La struttura del centro servizi è prevalentemente in acciaio. Le uniche parti in calcestruzzo armato sono i setti degli ascensori e dei vani tecnici e 6 pilastri posti in corrispondenza dell'auditorium.

I pavimenti sono costituiti da una soletta in cemento armato gettato su una lamiera grecata e da finiture di vario tipo in base alla zona dell'edificio. Tutti gli impianti passano nei controsoffitti e arrivano ai vari livelli del centro servizi tramite i cavedi tecnici posti affianco agli ascensori.

Ad entrambi i lati più corti del centro servizi vi sono le scale antincendio, anch'esse rivestite dalle stesse lamiera modulari che ricoprono il corpo di fabbrica centrale.

## 1.2 Obiettivi da raggiungere

L'obiettivo prefissato è quello di poter esplorare il centro servizi della sede unica della Regione Piemonte e, tramite l'utilizzo della realtà aumentata, visualizzare sullo schermo di un dispositivo mobile informazioni riguardanti la posizione degli impianti non visibili (ad esempio quelli che passano dietro i controsoffitti) e le informazioni relative alla manutenzione di ogni singolo elemento.

Quest'applicazione sviluppata per smartphone e tablet è pensata per poter camminare ed esplorare fisicamente l'edificio e, tramite la realtà aumentata, fare delle indagini in loco non invasive. L'applicazione sviluppata nel presente lavoro trova utilità anche per verificare, durante le fasi di cantiere, che le lavorazioni avvengano secondo quanto definito in fase di progetto.

Inoltre si è scelto di utilizzare la realtà aumentata piuttosto che quella virtuale poiché questa applicazione è pensata per poter essere utilizzata eventualmente anche in realtà lavorative piccole: si è pensato che fosse più facile sostenere una spesa economica di un iPad piuttosto che quella di un visore che ha decisamente un costo più alto e un campo di applicazione più ristretto.

Oltre alla visualizzazione delle informazioni si procederà anche all'esportazione di questi su un database in cloud per poi reimportarli il Revit in un secondo momento.



## 2 METODOLOGIA

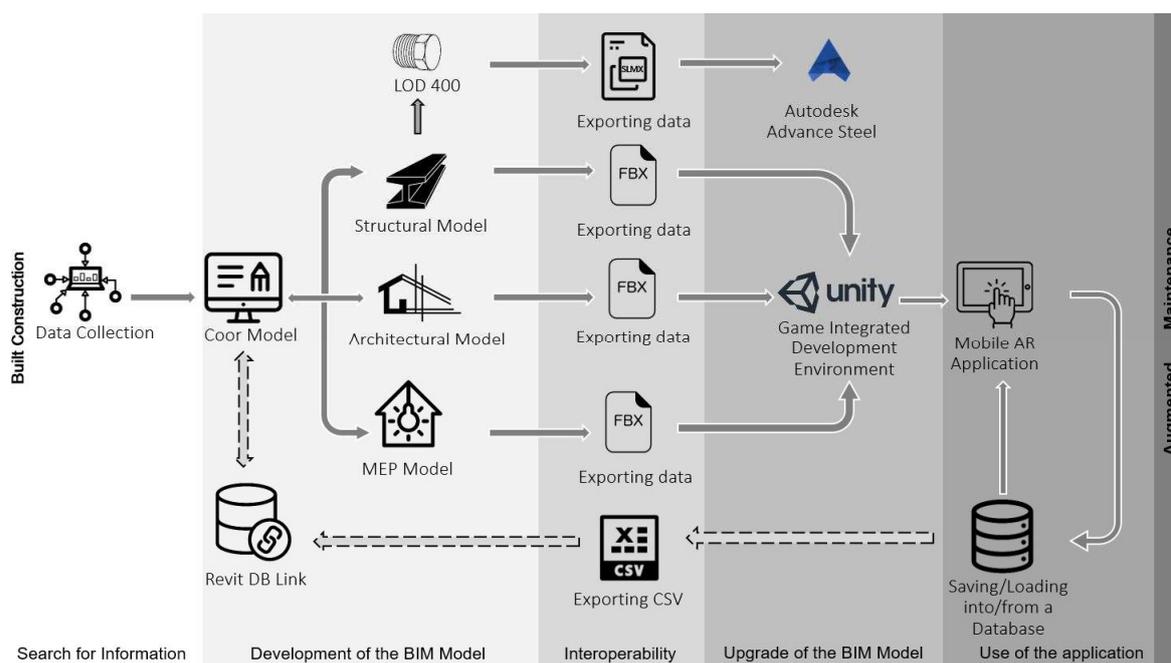


Figura 3: Workflow

Il processo di realizzazione del presente lavoro è suddivisibile in sei macro fasi:

- La raccolta delle informazioni dai file CAD forniti dalla regione
- La realizzazione del modello BIM
- La fase di esportazione ed interoperabilità con altri software
- Lo sviluppo di alcuni aspetti del modello BIM tramite l'utilizzo di altri programmi
- L'utilizzo della realtà aumentata per registrare le manutenzioni effettuate
- L'importazione dei dati ottenuti con l'applicazione all'interno di Revit

Le prime due fasi sono in parte sovrapposte in quanto, come si vedrà in seguito, i file CAD sono stati caricati all'interno del modello BIM come traccia da seguire per la modellazione. Il Coor Model rappresenta il file centrale del modello BIM al quale vengono linkati i file relativi allo strutturale, agli impianti e all'architettonico. Il modello strutturale, architettonico e antincendio sono stati la base di partenza per la modellazione. Per quanto riguarda il modello strutturale è stato aumentato il LOD. Inoltre sono state fatte delle prove di interoperabilità con il software Advance Steel per cercare di estrapolare quante più informazioni possibili relativamente ai giunti. Per quanto riguarda la parte impiantistica è stato implementato il modello dell'impianto elettrico. Sono stati modellati gli elementi appartenenti ad esso e, per quanto riguarda l'illuminazione, sono state inserite le informazioni relative al voltaggio delle singole luci e i circuiti a cui sono collegate e i quadri a cui quest'ultimi sono collegati.

L'ultimo sviluppo del modello BIM è stato quello di creare un'applicazione che sfruttasse la realtà aumentata per poter registrare le date in cui vengono effettuate le manutenzioni ai singoli elementi degli impianti. Inoltre, dato che il Processo BIM non è un processo a senso unico ma è ciclico, ovvero le informazioni ottenute ritornano al punto di partenza, è stato studiato un metodo per inviare le date registrate ad un database online direttamente dall'applicazione. In un secondo momento è possibile scaricare il database e, tramite il Revit DB Link, importare le informazioni aggiornate all'interno del modello. Infine è stato creato un abaco delle manutenzioni dal quale è possibile visualizzare quando sono state fatte le manutenzioni ai singoli elementi e quando è la relativa scadenza.

Per svolgere questo lavoro sono stati utilizzati gli hardware e i software di seguito elencati.

### Hardware Utilizzati

Nome Dispositivo	Specifiche Tecniche	Descrizione
	<p>Sistema Operativo: Windows 10 Home</p> <p>Processore: Intel® Core™ i7-7700HQ CPU 2.80GHz</p> <p>RAM: 16 GB</p> <p>GPU: NVIDIA GeForce GTX 1060 with Max-Q Design</p>	<p>Dispositivo utilizzato per creare il modello in Revit e per fare le varie esportazioni del modello.</p>
	<p>Sistema operativo: macOS High Sierra 10.13.4</p> <p>Processore: 2.9 GHz Intel® Core™ i7</p> <p>RAM: 8Gb DDR3</p> <p>GPU: Intel HD Graphics 4000 1536 Mb</p>	<p>Dispositivo utilizzato per creare il progetto in Unity3D e per installare l'applicazione su dispositivi mobili Apple tramite Xcode.</p>

<p><b>iPad Pro 9.7" 32 Gb</b></p> 	<p>Schermo: 9,7" con True Tone display.  CPU: Apple A9X a 64 bit con clock a 2,16 GHz.  RAM: 2 Gb.  Fotocamera posteriore: 12 megapixel e video in 4K</p>	<p>Dispositivo utilizzato per testare l'applicazione creata con Unity3D.</p>
<p><b>iPhone 7 128 Gb</b></p> 	<p>Sistema Operativo: iOS 11.3  Schermo: 4,7" Retina HD, con risoluzione da 1.334 x 750 pixel e densità da 326 ppi.  CPU: Apple A10Fusion quad-core.  RAM: 2 Gb.  Fotocamera posteriore: 12 megapixel con OIS, f/1.8 e f.</p>	<p>Dispositivo utilizzato per testare l'applicazione creata con Unity3D.</p>

*Figura 4: Hardware utilizzati*

## Software Utilizzati

Nome Software	Specifiche Tecniche	Descrizione
 <b>AUTODESK® REVIT®</b>	Autodesk Revit 2017 Sistema Operativo: Windows 10	Programma utilizzato per la modellazione del centro servizi.
	Unity3D 2017.4.0f1 Sistema Operativo: Macintosh	Programma utilizzato per la creazione dell'applicazione (Gratis)
 <b>Xcode</b>	Xcode 9.3 Sistema Operativo: Macintosh	Programma utilizzato per installare l'applicazione su dispositivi mobili Apple (Gratis)
 <b>vuforia™</b>	Vuforia Sistema Operativo: Macintosh	Plug-in di Unity3D per creare app in AR basate su marker 2D/3D (Gratis)
	ARKit 1.5 Sistema Operativo: Macintosh	Applicazione usata in parallelo a Unity3D per la creazione dell'app in AR basata su marker per ambiente (Gratis)

*Figura 5: Software utilizzati*



## 3 IL BIM

### 3.1 Cos'è il BIM?

Il BIM (Building Information Modeling) è una metodologia di progettazione basata sull'interscambio di informazioni fra i vari attori che partecipano alla realizzazione di un progetto mediante l'uso di file e software interoperabili tra loro.

Il BIM in Italia non è ancora molto diffuso, a differenza dei paesi anglosassoni dove ormai lavorare con questa metodologia è quasi la normalità.

Grazie al BIM è possibile inserire oltre alle informazioni geometriche degli elementi, anche informazioni alfanumeriche relative all'elemento stesso. Alla base del BIM c'è lo scambio di informazioni. Ogni singolo attore non lavora più in maniera indipendente dalle altre figure professionali che prendono parte al progetto, ma vi è un costante scambio di informazioni, attinte da un unico database centrale. Altro fondamento del BIM è l'interoperabilità. Grazie alle potenzialità del BIM è possibile lavorare sul medesimo progetto utilizzando più software che si scambiano informazioni e dati per poter analizzare il medesimo modello sotto diversi aspetti quali, ad esempio, strutturale, energetico, facility management ecc, che porta ad una notevole riduzione degli errori in fase di progettazione.

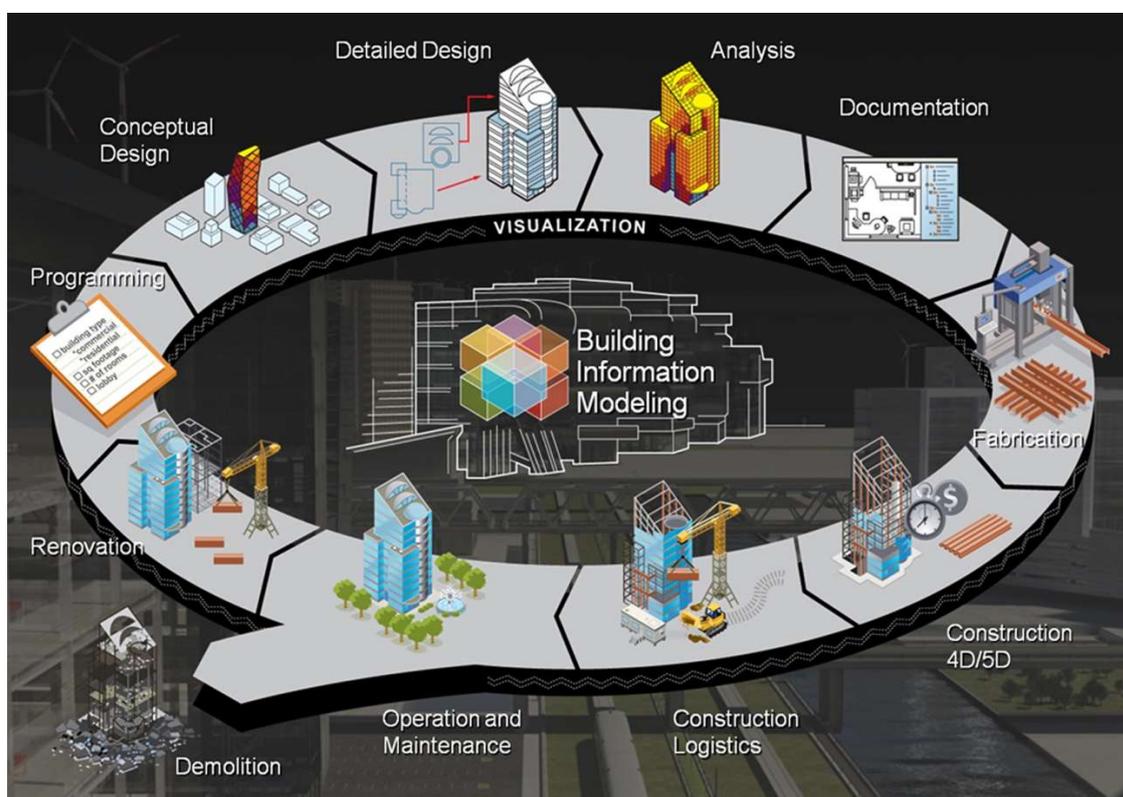


Figura 6: Workflow della progettazione BIM – Fonte: [http://www.sperastudio.it/?page\\_id=2658](http://www.sperastudio.it/?page_id=2658)

## 3.2 Le curve di MacLeamy

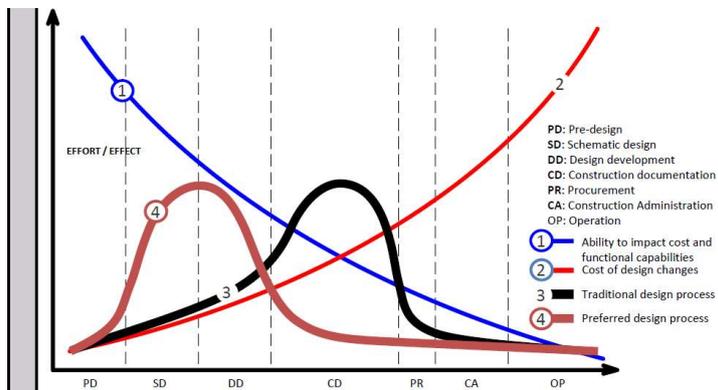


Figura 7: Curve di MacLeamy – Fonte: <http://www.bis-lab.eu>

Le curve di MacLeamy sono una pratica rappresentazione di come l'adozione del BIM porti notevoli vantaggi, sia allo studio di progettazione, che all'impresa che esegue i lavori, che al committente stesso.

Si può notare innanzitutto come il costo delle varianti in corso

d'opera aumentino esponenzialmente all'avanzare del processo di progettazione.

La curva 3 e la curva 4 rappresentano rispettivamente i costi della progettazione tradizionale e di quella adottando la metodologia BIM durante le varie fasi progettuali. Si può notare come nella fase iniziale i costi della progettazione BIM siano notevolmente più alti di quella tradizionale. Questo è dovuto alla pianificazione e al settaggio di tutti i parametri del progetto. Una volta avviata la progettazione però i costi diminuiscono drasticamente rispetto alla progettazione tradizionale fino ad azzerarsi quasi completamente nella fase "CD".

Questo perché si ha un controllo totale sul progetto e sulle informazioni dei singoli elementi e ciò permette di incorrere in errori più difficilmente rispetto alla progettazione tradizionale. Uno dei vantaggi è ad esempio quello di poter eseguire una "clash detection" già in fase progettuale, in modo tale da individuare interferenze tra elementi prima che il cantiere inizi e quindi di poter apportare modifiche quando è ancora possibile farlo senza dover sopportare costi troppo elevati.



## 4 AUMENTO DI LOD

### 4.1 Cos'è il LOD

Il LOD (Level of Detail) è il grado di dettaglio con cui un elemento del modello viene modellato. Esistono normative sia americane che italiane che definiscono le caratteristiche di ogni singolo LOD in base al tipo di elemento considerato.

La normativa americana è la “LOD Spec 2017 Part I”.

I livelli di LOD riportati dalla norma sono:

- LOD 100: l'elemento è rappresentato in maniera generica o con un simbolo ma non in maniera soddisfacente;
- LOD 200: l'elemento è rappresentato, con dimensioni, posizione e orientamento generico;
- LOD 300: l'elemento è rappresentato con dimensioni, posizione e orientamento corretto;
- LOD 350: l'elemento è rappresentato con dimensioni, posizione e orientamento corretto e si interfaccia correttamente con gli altri elementi del modello;
- LOD 400: l'elemento è rappresentato con dimensioni, posizione e orientamento corretto, si interfaccia correttamente con gli altri elementi del modello e ci sono dettagli sul montaggio;
- LOD 500: l'elemento rispecchia fedelmente la realtà.

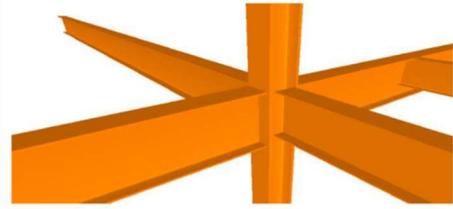
La norma indica per ogni tipo di elemento costruttivo, le varie proprietà che caratterizzano l'elemento ad ogni singolo LOD.

In particolare, per gli elementi strutturali in acciaio la norma indica le seguenti caratteristiche:

300

Element modeling to include:

- Specific sizes of main horizontal structural members modeled per defined structural grid with correct orientation, slope and elevation
- 



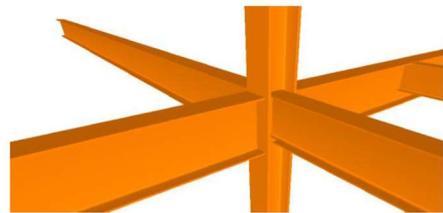
28 B1010.10-LOD-300 Floor Structural Frame (Steel Framing Beams)

Figura 8: Estratto dalla Norma LOD Spec 2017 Part I - LOD 300

350

Element modeling to include:

- Actual elevations and location of member connections
- Main elements of typical connections applied to all structural steel connections such as base plates, gusset plates, anchor rods, etc.
- Any miscellaneous steel members with correct orientation
- Any steel structure reinforcement such as web stiffeners, sleeve penetrations, etc.



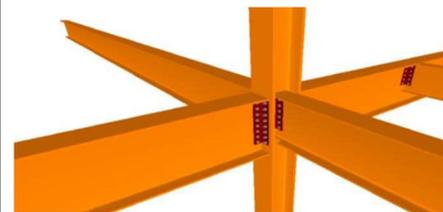
29 B1010.10-LOD-350 Floor Structural Frame (Steel Framing Beams)

Figura 9: Estratto dalla Norma LOD Spec 2017 Part I - LOD 350

400

Element modeling to include:

- Welds
- Coping of members
- Bent plates, cap plates, etc.
- Bolts, washers, nuts, etc.
- All assembly elements



30 B1010.10-LOD-400 Floor Structural Frame (Steel Framing Beams)

Figura 10: Estratto dalla Norma LOD Spec 2017 Part I - LOD 400

La normativa italiana che fa riferimento ai LOD è la UNI 11337-4:2017.

La presente normativa fa una distinzione fondamentale tra LOD, LOG e LOI.

LOD, livello di sviluppo degli oggetti digitali, è composto dai LOG, livello di sviluppo degli oggetti – attributi geometrici, e dai LOI, livello di sviluppo degli oggetti – attributi informativi.

Nella fattispecie, la normativa dà la seguente descrizione dei singoli LOD:

- LOD A: le entità sono rappresentate graficamente attraverso un sistema geometrico simbolico o una raffigurazione di genere presa a riferimento senza vincolo di geometria. Le caratteristiche quantitative e qualitative sono indicative.
- LOD B: le entità sono virtualizzate graficamente come un sistema geometrico generico o una geometria d'ingombro. Le caratteristiche qualitative e quantitative sono approssimate.
- LOD C: Le entità sono virtualizzate graficamente come un sistema geometrico definito. Le caratteristiche qualitative e quantitative sono definite in via generica entro e nel rispetto dei limiti della legislazione vigente e delle norme tecniche di riferimento e riferibili ad una pluralità di entità similari.
- LOD D: le entità sono virtualizzate graficamente come un sistema geometrico dettagliato. Le caratteristiche qualitative e quantitative sono specifiche di una pluralità definita di prodotti similari. È definita l'interfaccia con altri sistemi specifici di costruzione, compresi gli ingombri approssimati di manovra e manutenzione.
- LOD E: le entità sono virtualizzate graficamente come uno specifico sistema geometrico specifico. Le caratteristiche quantitative e qualitative sono specifiche di un singolo sistema produttivo legato al prodotto definito. È definito il livello di dettaglio relativo alla fabbricazione, l'assemblaggio, e l'installazione compresi gli specifici ingombri di manovra e manutenzione.
- LOD F: gli oggetti esprimono la virtualizzazione verificata sul luogo dello specifico sistema produttivo eseguito/costruito. Le caratteristiche quantitative e qualitative sono quelle specifiche del singolo sistema produttivo del prodotto posato e installato. Sono definiti per ogni singolo prodotto gli interventi di gestione, manutenzione e/o riparazione e sostituzione da eseguirsi lungo tutto il ciclo di vita dell'opera.

- LOD G: gli oggetti esprimono la virtualizzazione aggiornata dello stato di fatto di una entità in un tempo definito. Rappresentazione storicizzata dello scorrere della vita utile di uno specifico sistema produttivo aggiornato rispetto a quanto originariamente eseguito/costruito e installato. Sono definiti per ogni singolo prodotto gli interventi di gestione, manutenzione e/o riparazione e sostituzione da eseguirsi lungo tutto il ciclo di vita dell'opera.

LOD A	LOD B	LOD C	LOD D	LOD E	LOD F	LOD G
<b>Geometria</b> Elemento strutturale lineare orizzontale o pseudo-orizzontale rappresentato mediante un simbolo 2D.	<b>Geometria</b> Elemento strutturale lineare orizzontale o pseudo-orizzontale rappresentato mediante un solido di estrusione abbozzato.	<b>Geometria</b> Elemento strutturale lineare orizzontale o pseudo-orizzontale rappresentato mediante un solido avente dimensioni calcolate secondo la normativa tecnica. Sono definiti i collegamenti tipici resistenti.	<b>Geometria</b> Elemento strutturale lineare orizzontale o pseudo-orizzontale rappresentato mediante un solido avente dimensioni pari alle dimensioni reali. Sono modellate tutte le piastre e gli irrigidimenti relativi alla trave.	<b>Geometria</b> Elemento strutturale lineare orizzontale o pseudo-orizzontale rappresentato mediante un solido avente dimensioni pari alle dimensioni reali. Sono modellati anche tutti gli elementi necessari per la produzione quali bulloni e saldature. Anche gli assemblaggi sono definiti a modello.	<b>Geometria</b> Come LOD E (rilevo di quanto eseguito).	<b>Geometria</b> Nuovi interventi: Come LOD F (con aggiornamenti) Manutenzione e gestione su elementi esistenti: Come LOD C o D (a partire da).
<b>Oggetto</b> Simboli grafici 2D	<b>Oggetto</b> Solido 3D	<b>Oggetto</b> Solido 3D complesso	<b>Oggetto</b> Solidi 3D complessi	<b>Oggetto</b> Solidi 3D complessi	<b>Oggetto</b> Solidi 3D complessi	<b>Oggetto</b> Solidi 3D complessi
<b>Caratteristiche</b> - Posizionamento di massima	<b>Caratteristiche</b> - Materiali ipotizzabili - Sezioni ipotizzabili	<b>Caratteristiche</b> - Materiali da calcolo - Sezioni calcolate	<b>Caratteristiche</b> - Piastre 3D - Irrigidimenti 3D - Informazioni su bulloni e saldature	<b>Caratteristiche</b> - Bulloni 3D - Saldature 3D - Assemblaggi - Fasi di montaggio	<b>Caratteristiche</b> - Certificati di collaudo - Piano di manutenzione	<b>Caratteristiche</b> - Data di manutenzione/sostituzione - Soggetto manutentore - Tipologia di intervento

Figura 11: LOD per elementi strutturali (UNI 11337-4:2017)

Per la modellazione è stata seguita la normativa italiana in quanto, essendo l'edificio situato in Italia, è stato ritenuto opportuno seguire la normativa vigente sul territorio nazionale piuttosto che quella americana.

## 4.2 Applicazione al caso pratico

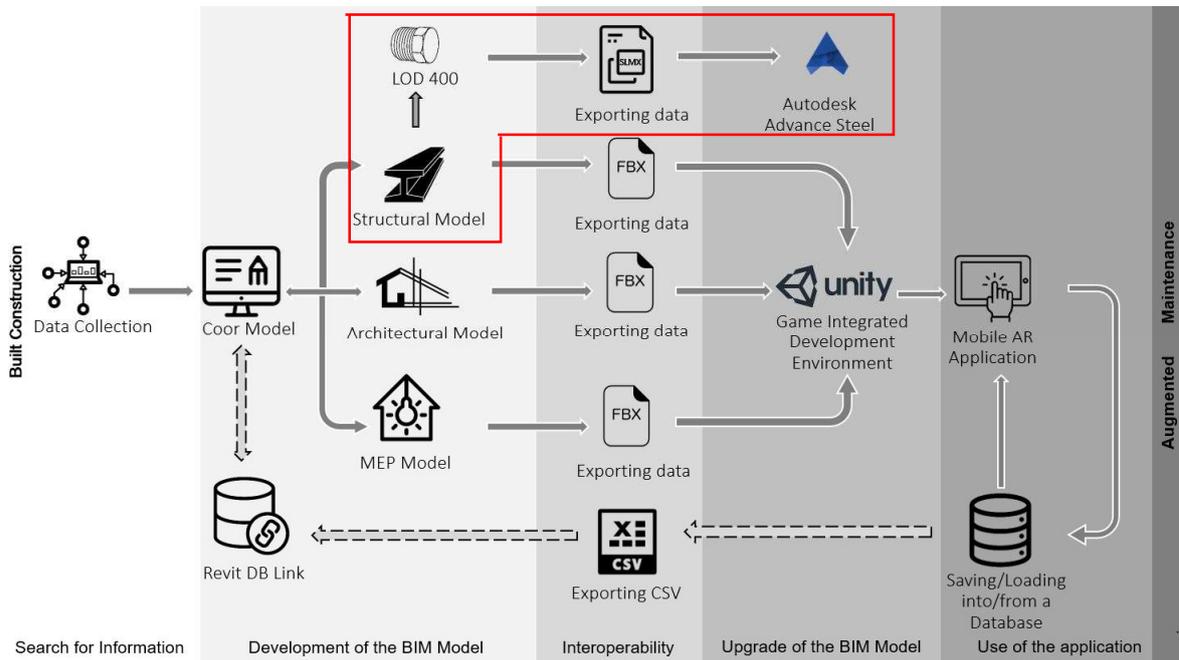


Figura 12: Posizione all'interno del workflow - Aumento di LOD

Il punto di partenza, come detto già in precedenza è stato lo strutturale modellato al LOG C (LOD 300), dunque tutte le travi e i pilastri, nonché i setti e le solette, erano stati modellati inserendo le corrette dimensioni, posizioni e orientamento. Mediante l'utilizzo del plug-in Advance Steel è stato possibile implementare i giunti tra gli elementi in acciaio in maniera quasi automatica.

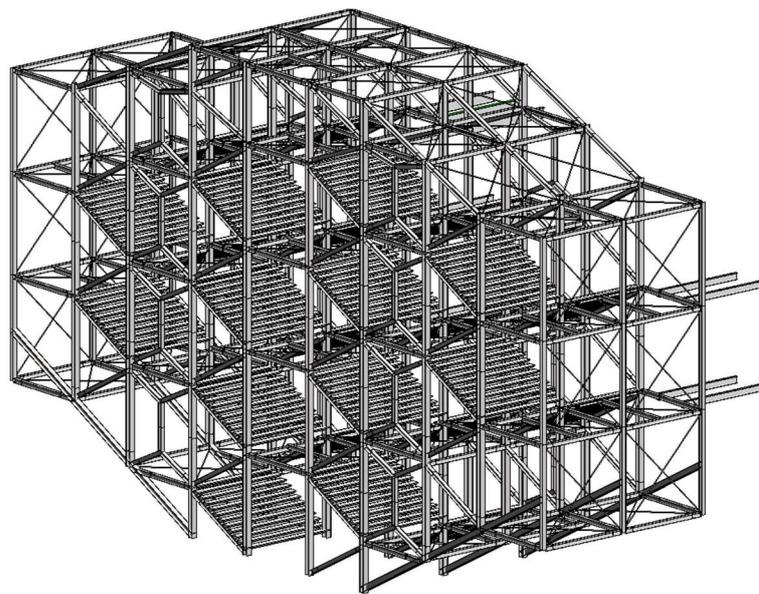


Figura 13: Scala antincendio - LOD C (LOG 300)

## 4.2.1 Impostazione delle famiglie

Innanzitutto è stato scaricato il software Autodesk Advance Steel e il rispettivo plug-in per Revit 2017. In Revit 2018 è stata implementata la possibilità di scaricare solo il plug-in senza dover scaricare anche il programma. Una volta scaricato il plug-in è stato necessario sostituire tutte le famiglie di travi inserite.

Il software consente la creazione di giunti solo su elementi strutturali appartenenti a famiglie “certificate”.

Nelle librerie Italiane queste non sono presenti, dunque si è scelto di utilizzare le famiglie appartenenti alle librerie tedesche le quali invece risultavano essere “certificate”.

Sul sito della Autodesk sono comunque elencate tutte le librerie che contengono famiglie certificate.

Germania	C:\ProgramData\Autodesk\RVT 2017\Libraries\Germany\Tragwerk Trager\Stahl
	C:\ProgramData\Autodesk\RVT 2017\Libraries\Germany\Tragwerk Stutzen\Stahl
	C:\ProgramData\Autodesk\RVT 2017\Libraries\German_INTL \Tragwerk Trager\Stahl\AISC 14.1
	C:\ProgramData\Autodesk\RVT 2017\Libraries\German_INTL \Tragwerk Stutzen\Stahl\AISC 14.1

*Figura 14: Percorso famiglie tedesche certificate*

## 4.2.2 Creazione dei giunti

I giunti possono essere creati sia da Revit che da Advance Steel. In questo caso sono stati creati in Revit poiché tutti i giunti necessari erano presenti in quelli disponibili. Inoltre sarebbe stato più complessa la modellazione su Advance steel in quanto sarebbe stato necessario importare nuovamente tutti i file CAD all'interno del modello esportato con il rischio di commettere errori di allineamento.

Prima di procedere alla creazione dei giunti, si sono importate le tipologie necessarie. Importando solo le tipologie necessarie di giunti si alleggerisce il modello.

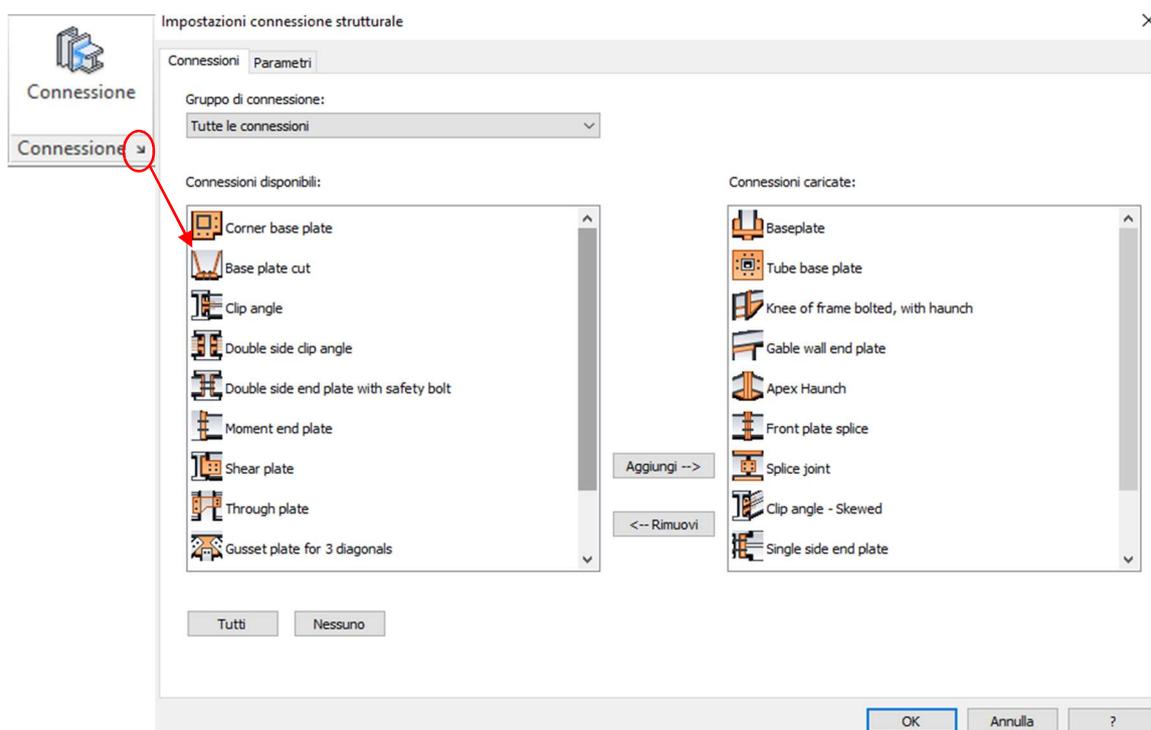


Figura 15: Elenco connessioni importate all'interno del modello strutturale

Una volta importate le famiglie a seconda del giunto da inserire, si selezionano gli elementi desiderati e, cliccando su “Connessione”, si sceglie la tipologia di giunto da inserire.

Se questo procedimento viene fatto con famiglie non certificate o vengono selezionati elementi non previsti per quel tipo di giunto, il programma darà come output un messaggio di errore.

Inserito il giunto, il programma inserisce delle misure casuali all'elemento, bisogna dunque andare a modificare manualmente i vari parametri del giunto.

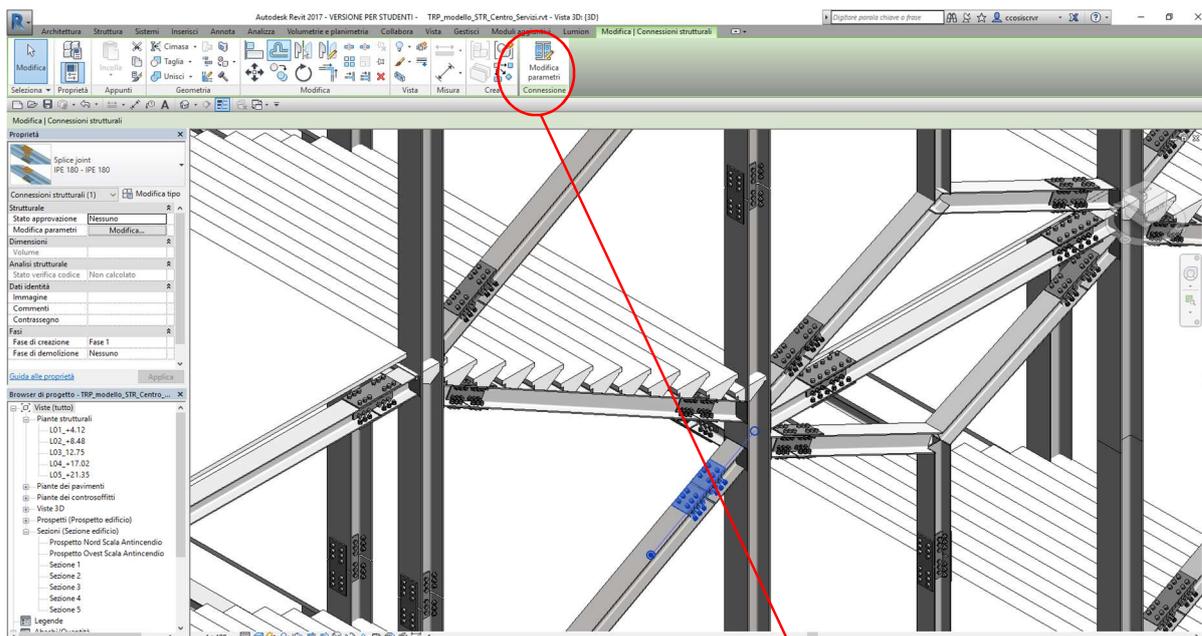


Figura 17: Giunti Scale Antincendio

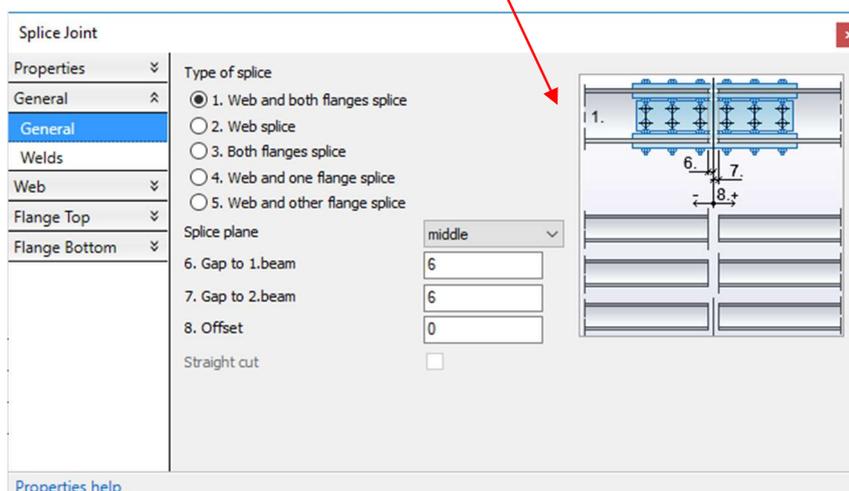


Figura 16: Centro Giunti – Esempio di “Splice Joint” di un controvento

Tutte le tipologie di giunti utilizzate in questo progetto sono risultate altamente dettagliate e vi sono numerosi parametri utilizzabili per ottenere la forma voluta.

Oltre alla modellazione puramente geometrica, il plug-in di Advance Steel permette di fare calcoli strutturali sui giunti. Nel caso in cui un giunto non dovesse sopportare i carichi a cui è sottoposto, il programma dice su quale parte del giunto andare ad intervenire per risolvere il problema. È possibile ottenere pertanto un report dettagliato (Allegato 12.1).

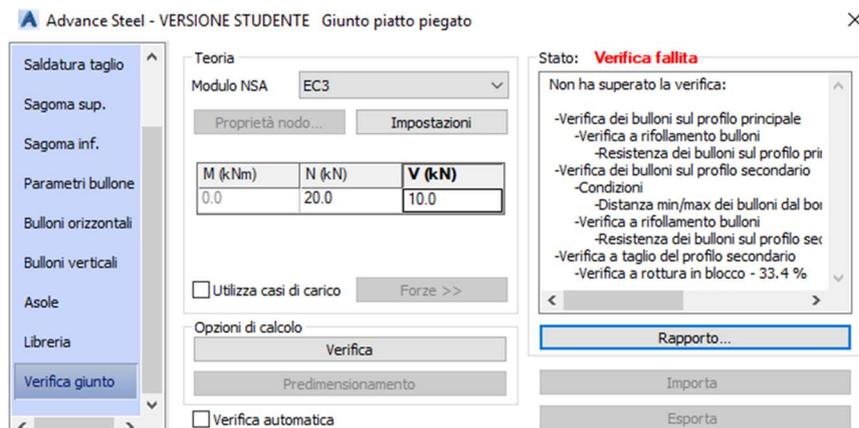


Figura 18: Risultato Giunti (Valori Fittizi)

In Revit 2017 non vi sono però tutti i giunti che effettivamente sono necessari. Per ovviare a questo problema si è esportato il modello in Advance Steel attraverso la sezione dedicata nella scheda plug-in.

Nel presente progetto non è stata comunque riscontrata l'assenza di tipi di giunti necessari all'interno di quelli presenti in Revit.

#### 4.2.3 Modellazione in Advance Steel

Una volta esportato il file da Revit si procede all'importazione dello stesso in Advance Steel. Va sottolineato che questa procedura va fatta una sola volta in quanto successivamente basta scegliere l'opzione "Synchronization" per far caricare a Revit solo le modifiche che sono state fatte in Advance Steel e viceversa. Una volta importato il modello in Advance Steel, si possono inserire i giunti seguendo una procedura analoga a quella vista in Revit.

Il vantaggio di inserire i giunti in Advance Steel rispetto a Revit 2017 è che prima di andare a selezionare il singolo giunto, si può leggere una breve descrizione del giunto stesso nella quale è indicato quali elementi selezionare, l'ordine di selezione, ecc..

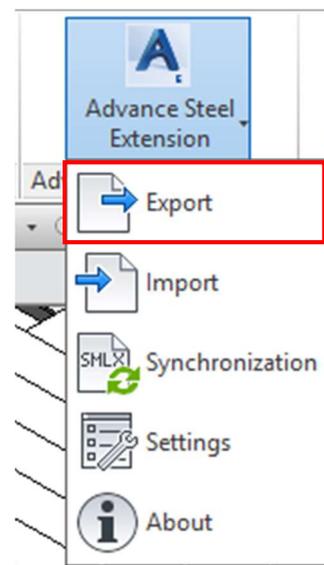


Figura 19: Esportazione da Revit in Advance Steel

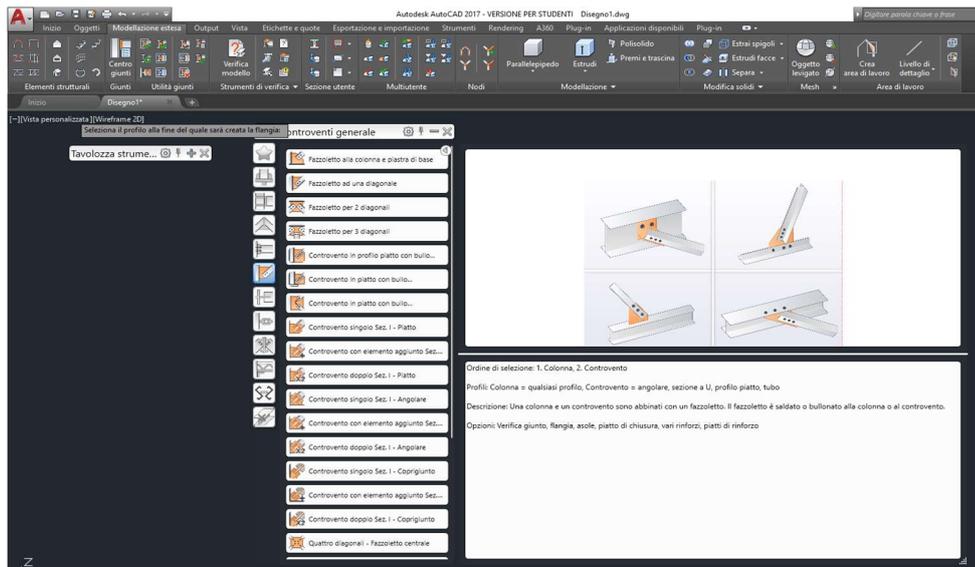


Figura 20: Centro Giunti in Advance Steel

Il vantaggio di usare Revit per la modellazione del telaio strutturale è la sua immediatezza e semplicità d'utilizzo, cosa che in Advance Steel, facendo dei test separati dal presente modello, non è stata riscontrata. Per quanto riguarda invece la modellazione dei giunti, questa risulta più immediata e semplice in Advance Steel. L'unico problema riscontrato è stato che se si utilizzano tipi di giunti non presenti in Revit, durante la sincronizzazione il programma crasha perché non riesce a riconoscere il tipo di giunto diverso. Questa è stata una delle motivazioni che ha portato alla scelta di utilizzare Revit per la modellazione.

Di seguito vengono riportati delle immagini del risultato finale della modellazione del modello strutturale.

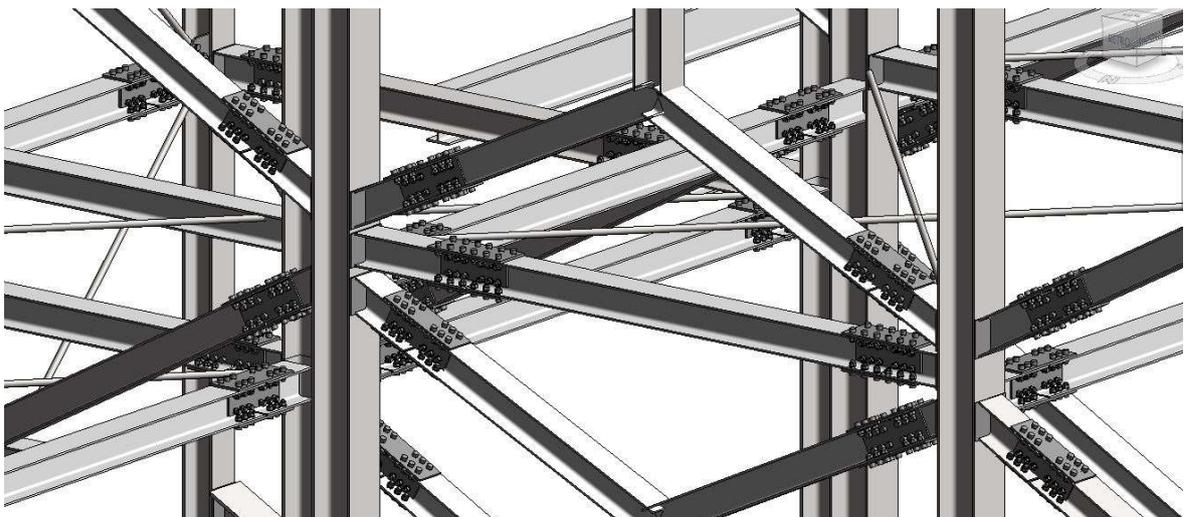


Figura 21: Dettaglio dei giunti dei controventi della scala antincendio – LOG E (LOD 400)

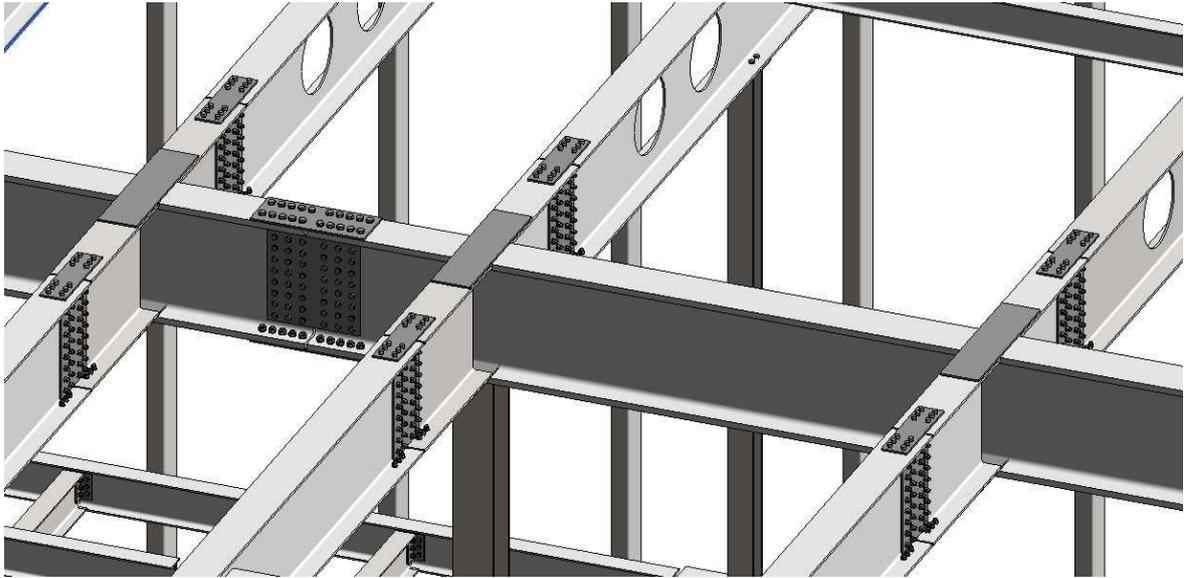


Figura 23: Dettaglio dei giunti delle travi poste in copertura – LOG E (LOD 400)

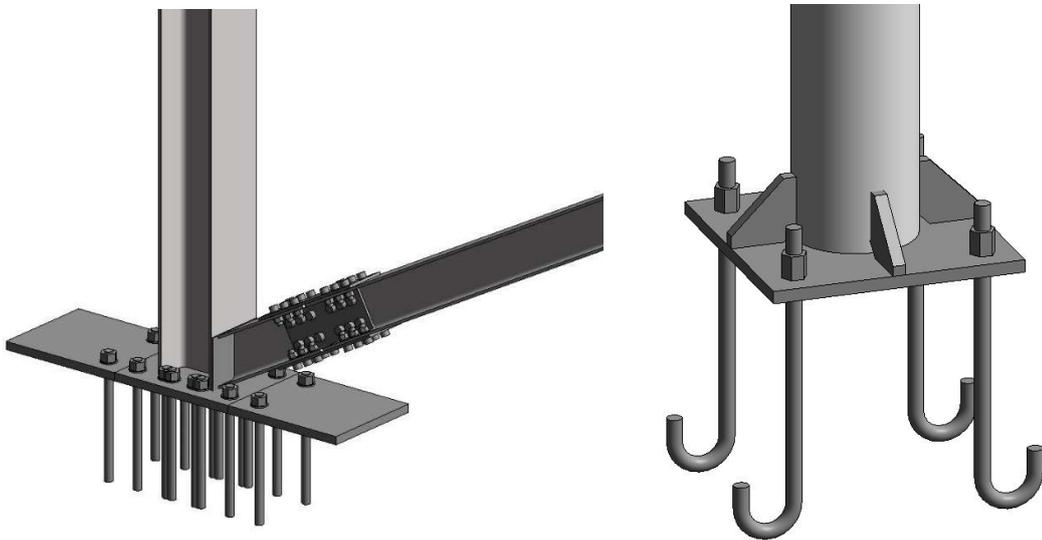


Figura 22: Dettaglio delle fondazioni. A sinistra fondazioni della scala antincendio a destra dell'asilo nido – LOG E (LOD 400)

## 4.2.4 Abaco delle connessioni strutturali in Revit

Finita la modellazione di tutti i giunti in acciaio, è stato creato un abaco delle connessioni strutturali nell'ottica di poter creare, in futuro, un computo metrico.

La prima criticità riscontrata è che quando si va a inserire un giunto bullonato su una trave, il software non detrae il materiale del foro dalla trave.

Per fare questa verifica è stato creato un nuovo progetto il quale conteneva 2 travi identiche. Sulla prima è stato applicato un giunto bullonato mentre sulla seconda no. Andando a creare un abaco dei materiali si è notato che entrambe le travi presentavano lo stesso volume, ciò ha portato alla conclusione che Revit non detrae i materiali delle bullonature.

Ciò porta a un errore nella computazione dei materiali, anche se questo risultando essere molto piccolo, che è stato deciso di trascurare.

<Abaco delle connessioni strutturali>						
A	B	C	D	E	F	G
Famiglia	Tipo	Conteggio	Bolt Size	Number Of Bolts	Costo	Livello
Baseplate	Baseplate	54			0.00	
Clip angle - Skewe	IPE 140	157				
Clip angle - Skewe	IPE 240 - IPE 240	4				
Clip angle - Skewe	IPE 240 - IPE 300	1				
Clip angle - Skewe	IPE 270 - HEB 400	114				
Clip angle - Skewe	IPE 270 - IPE 270	122				
Clip angle - Skewe	IPE 270 - IPE 720	162				
Clip angle - Skewe	IPE 300 - IPE 300 (	1				
Clip angle - Skewe	IPE 300 - IPE 400	4				
Clip angle - Skewe	IPE 400 - HEB 800	34				
Clip angle - Skewe	IPE 400 - IPE240	34				
Clip angle - Skewe	IPE 400 - IPE 270	14				
Clip angle - Skewe	IPE 400 - IPE 400	4				
Clip angle - Skewe	IPE 400 - IPE 720	28				
Front plate splice	Front plate splice	4				
Front plate splice	HEB 400 - HEB 400	75				
Front plate splice	HEB 400 - HEM 800	1				
Front plate splice	HEM 800 - HEM 80	6				
Front plate splice	IPE240 - IPE 240	30				
Front plate splice	IPE400 - IPE 400	31				
Front plate splice	IPE 160 - IPE 160	184				
Front plate splice	IPE 270 - IPE 270	51				
Front plate splice	IPE 720 - IPE 720	3				

Figura 24: Stralcio dell'abaco delle connessioni strutturali in Revit

L'abaco è stato impostato raggruppando prima per "Famiglie" e successivamente per "Tipo" i vari elementi. L'intero Abaco è presente all'Allegato 12.2

Questo abaco non risulta soddisfacente in quanto si riesce ad estrapolare solo il conteggio di ogni tipo di famiglia di connessione strutturale.

Non si riescono invece ad ottenere le informazioni riguardanti il numero e la dimensione dei bulloni o il livello associato a una determinata connessione strutturale.

Questo problema è dovuto dal fatto che per Revit la singola connessione, seppur parametrizzabile, viene considerata un unico elemento, dunque non si riesce a selezionare il singolo bullone appartenente ad una determinata connessione strutturale.

Per ovviare a questo problema è stato creato un abaco delle connessioni strutturali in Advance Steel.

## 4.2.5 Abaco delle connessioni strutturali in Advance Steel

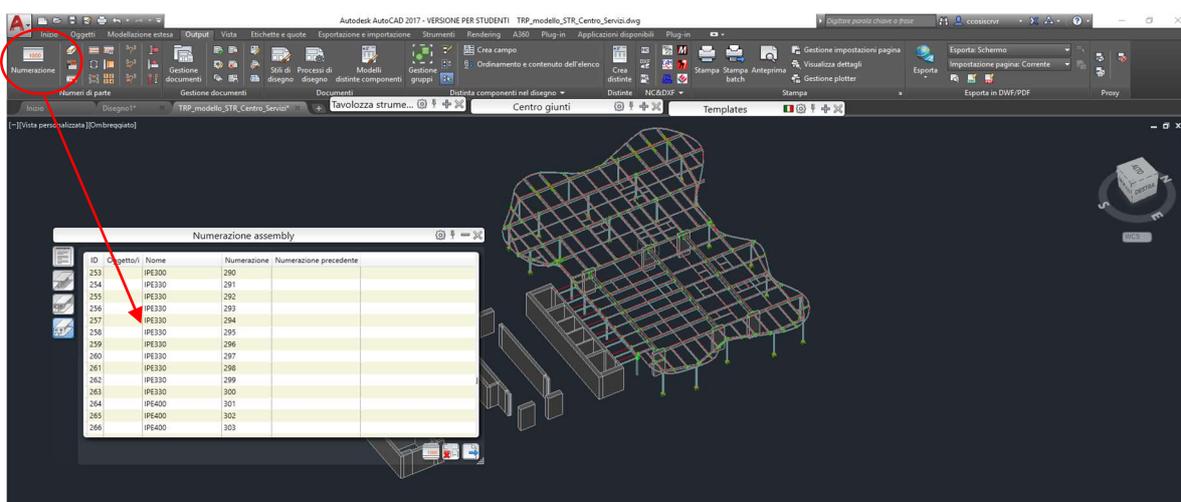


Figura 25: Numerazione Elementi

Per ottenere le informazioni sui bulloni è stata creata la distinta dei bulloni in Advance Steel. Per prima cosa sono stati numerati tutti gli elementi presenti nel progetto. Se non si facesse la numerazione degli elementi il programma non sarebbe poi in grado di riconoscere i giunti collegati a quei elementi.

Successivamente è stata creata la distinta dei bulloni attraverso la funzione “Crea Distinta” presente nella scheda “Distinte”.

La distinta completa è presente nell'allegato 12.3. Dato che il processo di creazione della distinta è molto lungo e pesante per i computer, è stata creata solo la distinta dei bulloni dell'asilo nido presente al piano terra. Se si volesse fare la distinta dell'intero edificio la procedura sarebbe la stessa.

È inoltre possibile creare la medesima distinta per i piatti delle connessioni strutturali.

La tipologia dei bulloni e il materiale sono puramente indicativi in quanto non si disponeva di informazioni dettagliate. Le dimensioni e la posizione delle singole connessioni strutturali sono state ricavate dalle varie piante e sezioni fornite dal dipartimento DISEG e dalla Regione Piemonte. È stato ritenuto utile fare la distinta dei bulloni in quanto è così possibile

ottenere il numero esatto di ogni singolo elemento dei vari giunti. Tramite questo numero esatto è possibile ad esempio creare un computo metrico molto preciso.

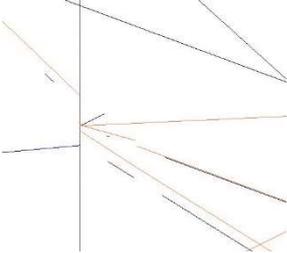
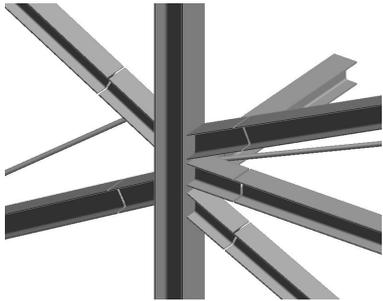
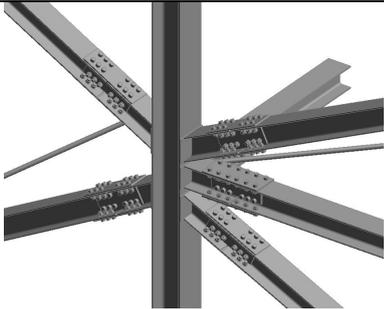
AUTODESK ADVANCE STEEL		Progetto:		Commessa:					
DISTINTA BULLONI		Cliente:		Disegnatore:		Data			
						11/04/2018			
Tipo codice	Qta	Materiale	Trattamento	Lunghezza (mm)	Tipo	Peso di pezzi (Kg/cad)	Peso Totale (kg)	Descrizione	
Bullone UNI 5712 M16x55	312	10.9	Zincato	55	UNI 5712	0,2	63,3		
Bullone UNI 5712 M16x50	99	10.9	Zincato	50	UNI 5712	0,2	19,3		
Bullone UNI 5712 M16x45	846	10.9	Zincato	45	UNI 5712	0,2	158,2		
Bullone UNI 5712 M16x40	14	10.9	Zincato	40	UNI 5712	0,2	2,5		
Bullone UNI 5712 M16x320	4	10.9	Zincato	320	UNI 5712	0,1	0,3		
Bullone UNI 5712 (DIN 6914) M20x75	36	8.8	Zincato	75	UNI 5712 (DIN	0,4	13,5		
Bullone UNI 5712 (DIN 6914) M16x70	102	8.8	Zincato	70	UNI 5712 (DIN	0,2	23		
Bullone UNI 5712 (DIN 6914) M16x50	80	8.8	Zincato	50	UNI 5712 (DIN	0,2	15,6		
Bullone ASTM A325 M19.05x69.85	6	10.9	Zincato	70	ASTM A325	0,4	2,5		
Bullone ASTM A325 M19.05x63.5	277	10.9	Zincato	64	ASTM A325	0,4	111,4		
Bullone ASTM A325 M19.05x57.15	475	10.9	Zincato	57	ASTM A325	0,4	184,3		
Bullone ASTM A325 M19.05x50.8	42	10.9	Zincato	51	ASTM A325	0,4	15,7		
Bullone ASTM A325 M19.05x44.45	29	10.9	Zincato	44	ASTM A325	0,4	10,4		

Figura 26: Estratto dalla distinta dei bulloni. Allegato 12.3

## 4.2.6 Confronto della modellazione degli elementi strutturali con la normativa

Di seguito viene riportato un confronto tra la normativa e la visualizzazione degli elementi in Revit in base al grado di dettaglio per correlare quest'ultimi ai LOD indicati nella UNI 11337-4:2017.

Viene riportato solo il LOG in quanto non sono state aggiunte informazioni relative agli elementi strutturali, non essendo queste state fornite.

			Geometria	Oggetto	Restituzione Grafica
LOG	LOG A	basso	Linea	Linea	
	LOG C	medio	Solido 3D semplificato. I raccordi vengono semplificati ad angoli retti. Non sono presenti giunti e saldature.	Solido 3D semplificato	
	LOG E	alto	Solido 3D complesso. Le dimensioni sono quelle reali, sono presenti i medesimi giunti e saldature presenti nella realtà.	Solido 3D complesso	



## 5 MODELLAZIONE MEP

### 5.1 Organizzazione dei file

L'intero progetto è strutturato con file linkati. Vi è un "Coor Model" al quale vengono linkati tutti i vari file nel quale il progetto è stato gerarchizzato.

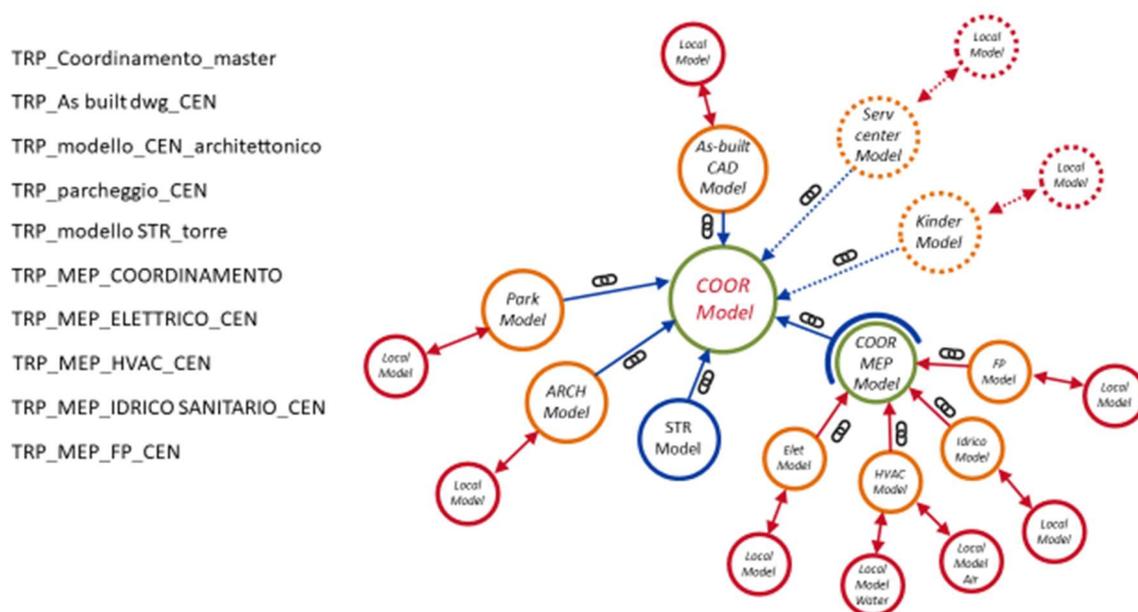


Figura 27: Schema organizzativo dei file

Nella fattispecie del centro servizi, vi è un "File di coordinamento" al quale vengono linkati i file delle discipline architettonica, strutturale e impiantistica, il quale a sua volta verrà linkato al "Coor Model"

I file degli impianti sono stati a loro volta suddivisi in Elettrico, Antincendio, HVAC e idricosanitario.

Al momento sono stati modellati solo l'impianto antincendio, modellato durante le ore di tirocinio, e l'elettrico, durante lo svolgimento di questa tesi.

Per la realizzazione dei file MEP (Mechanical, Electrical, Plumbing) si è utilizzato il "Work-sharing" per allinearsi alle scelte di progettazione utilizzate per la torre. Dell'impianto elettrico è stato modellato l'impianto antincendio (allarmi, rivelatori di fumo ecc..), l'impianto di illuminazione e l'impianto acustico.

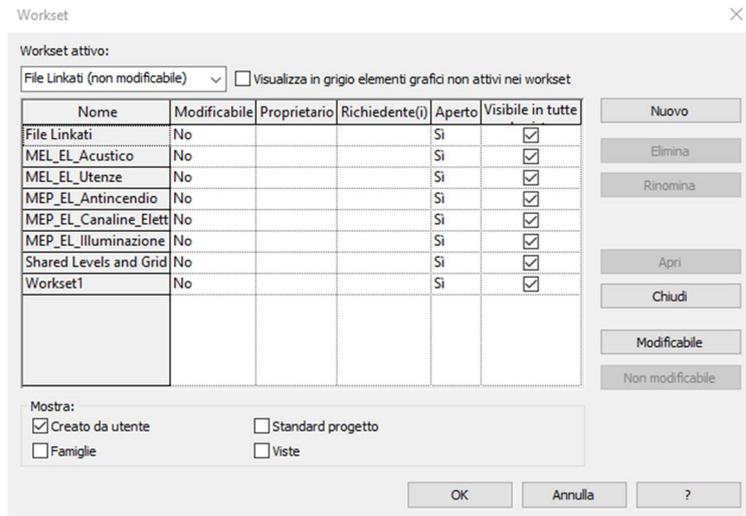


Figura 30: Workset utilizzati per l'impianto elettrico

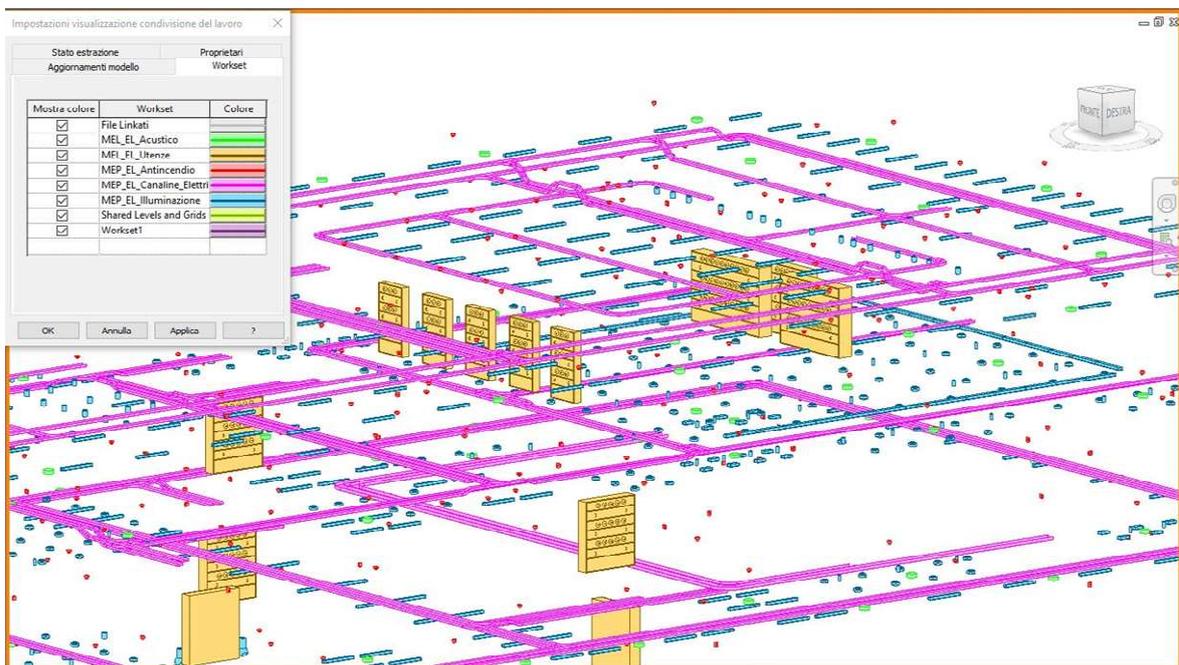


Figura 28: Visualizzazione 3D del modello con individuazione dei workset mediante diversi colori

## 5.2 La modellazione

### 5.2.1 L'inserimento delle famiglie

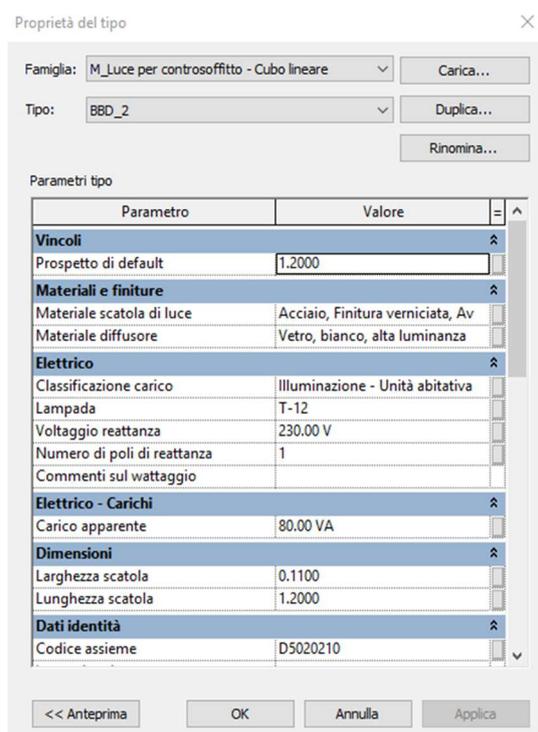


Figura 31: Proprietà di un tipo di luce a controsoffitto

Per prima cosa è stato linkato il file dello strutturale della torre.

Questo file è molto importante in quanto sono state acquisite le coordinate condivise di progetto che conteneva.

Una volta fatto ciò, è stato linkato il file contenente i dwg relativi all'impianto elettrico. Questo file, opportunamente allineato con la torre precedentemente importata, è servito come base per l'inserimento delle varie famiglie.

Un altro file che è stato linkato è stato quello dell'architettonico in quanto le famiglie inserite hanno come "host" i controsoffitti dunque serviva una base di riferimento per l'inserimento di questi elementi. Una volta fatto ciò è iniziato

l'inserimento delle famiglie nei relativi workset. Innanzi tutto sono stati creati i vari tipi di famiglie utilizzando la medesima nomenclatura riportata nei file dwg e nel file dei quadri elettrici.

Una volta inserite tutte le famiglie si è passato a collegare i singoli dispositivi di illuminazione ai relativi circuiti e, successivamente, ai relativi quadri elettrici.

### 5.2.2 La creazione dei circuiti elettrici

Il collegamento delle famiglie ai circuiti elettrici e ai relativi quadri è stato fatto solo per i dispositivi di illuminazione, in quanto non si disponeva della documentazione relativa ai circuiti dei sistemi di allarme antincendio, acustico e di sorveglianza.

Per quanto riguarda i dispositivi di illuminazione, invece, le informazioni sono state reperite dai file denominati "Quadri Centro Servizi.pdf" nel quale sono presenti

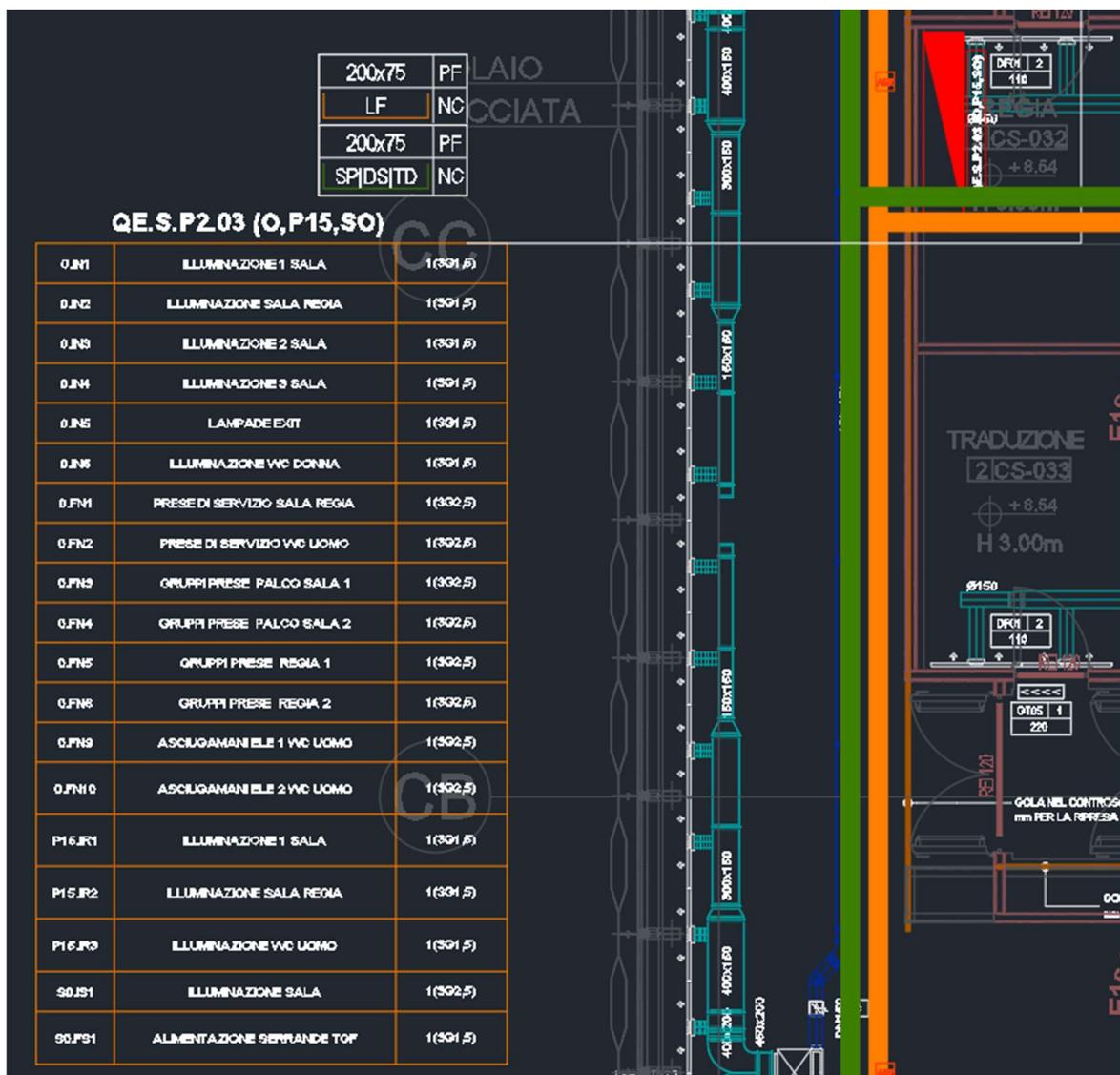


Figura 32: Estratto del file "PR-3-CEP-CSSZ1.dwg" – Elenco dei circuiti collegati al quadro elettrico

tutti gli schemi unifilari dei dispositivi di illuminazione e "PR-3-CEP-CSSZ1.dwg" nel quale vi è la pianta del secondo piano con indicati affianco ad ogni quadro elettrico, il nome del circuito, la destinazione d'uso del locale servito e il tipo di filo utilizzato per il circuito.

La nomenclatura utilizzata per i circuiti è suddivisa in:

- IN: Illuminazione normale
- IS: Illuminazione di sicurezza
- P: Illuminazione Privilegiata

- FN: Prese di corrente

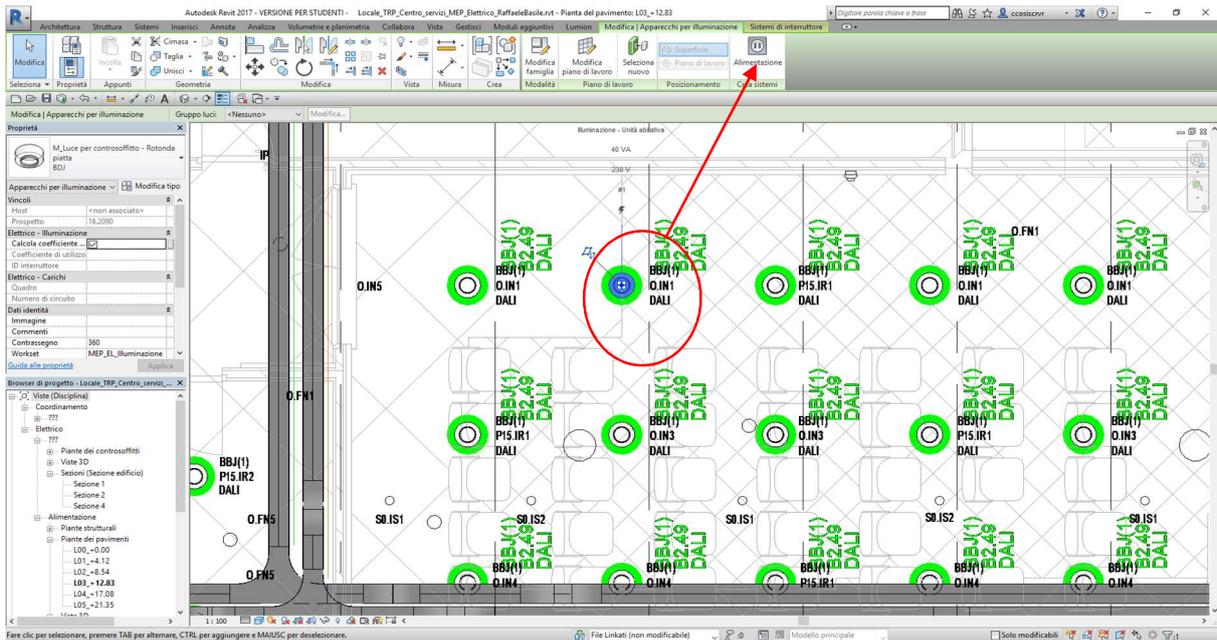


Figura 33: Creazione Circuito - Esempio di un aula Conferenza in ambiente di modellazione BIM (Revit)

Per prima cosa sono stati creati i circuiti ed inseriti i vari elementi appartenenti allo stesso. Per fare ciò bisogna cliccare sulla famiglia interessata e successivamente su Alimentazione. Così facendo viene creato in automatico un circuito contenente solo la famiglia selezionata e non collegato a nessun quadro.

Per aggiungere un nuovo elemento al circuito si deve selezionare il comando “Modifica Circuito”.

Tramite questa funzionalità è possibile aggiungere o rimuovere elementi dal circuito e selezionare il quadro a cui è collegato.

Quest’ultima opzione la si può fare anche cliccando sull’omonimo tasto presente alla destra di “Modifica Circuito”.

Una volta inseriti tutti gli elementi all'interno dei rispettivi circuiti, Revit organizza in automatico tutte le famiglie con una gerarchia dei componenti.

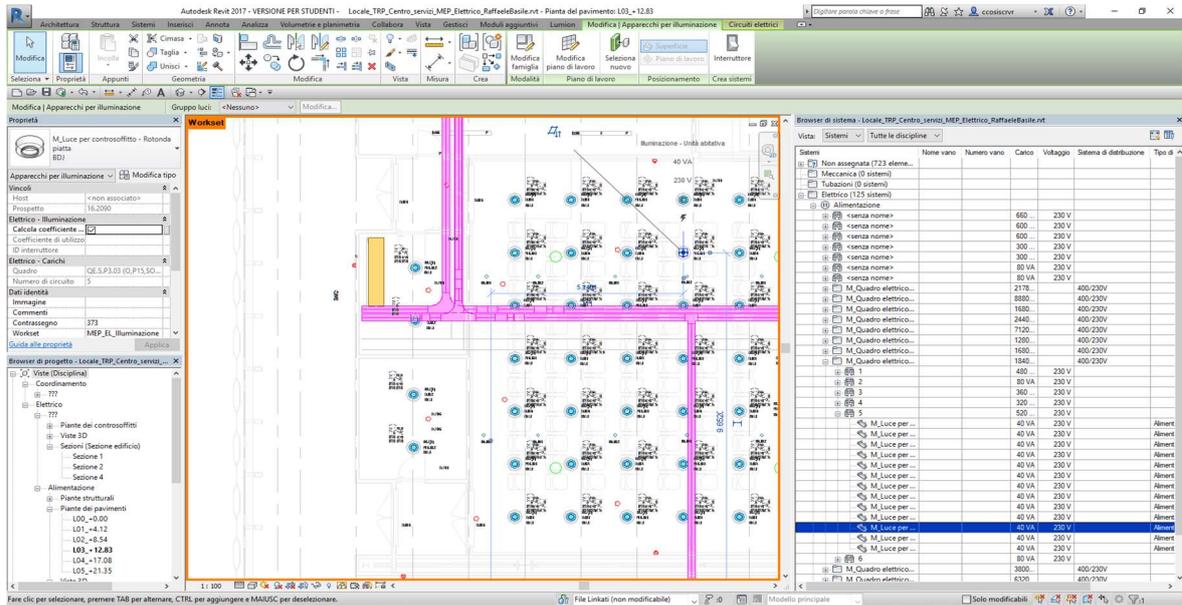


Figura 35: Gerarchia dei componenti

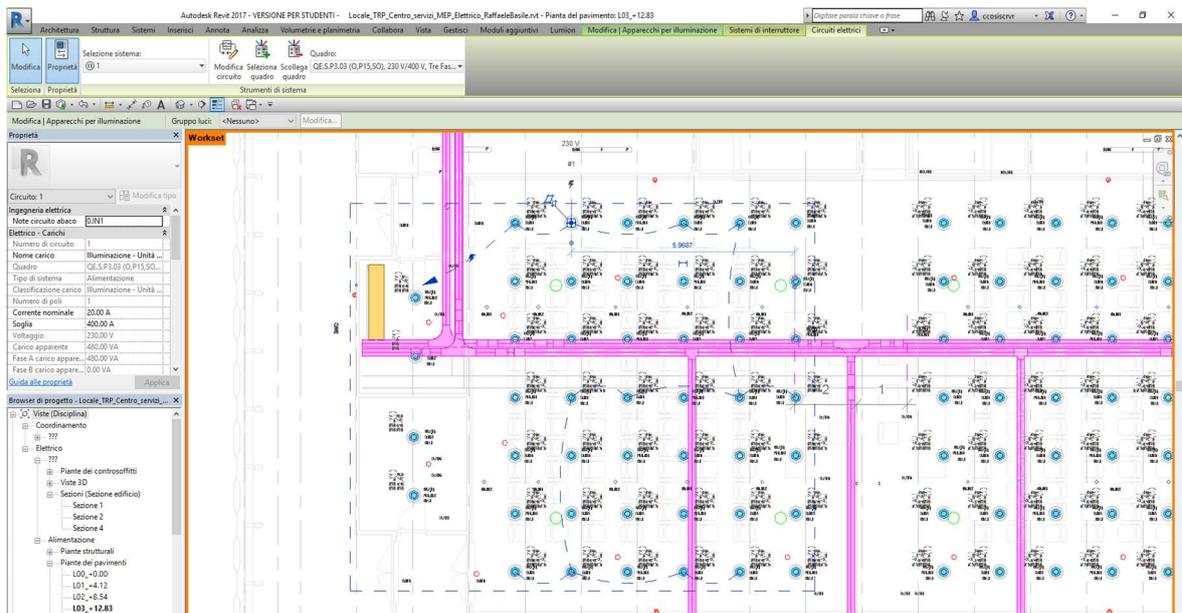


Figura 34: Schema logico circuito

Infatti, selezionando le singole famiglie direttamente dal progetto verranno individuate in maniera univoca all'interno del "browser di sistema".

Una volta creati i circuiti, è stato inserito in "Note circuito abaco" il nome che i progettisti avevano scelto per quel determinato circuito e in "Commenti" la destinazione d'uso del locale servito in modo tale da avere una visualizzazione più chiara all'interno dell'abaco.

Ciò è stato fatto poiché Revit non permette di rinominare i circuiti, ma gli assegna come nome un numero progressivo crescente. Questo metodo di numerazione dei circuiti non risulta molto chiaro e intuitivo.

### 5.2.3 Gli abachi dei circuiti elettrici

Una volta creati tutti i circuiti e collegati ai relativi quadri, si è passato a impostare l'abaco per poter visualizzare in modo chiaro e diretto i quadri elettrici e i circuiti collegati ad essi.

QE.S.P0.02 (O,P15,20), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
O.IN1	Illuminazione Sbarco Ascensori Liv +1		12	1
O.IN3	Illuminazione 1 Hall Liv +1		11	1
O.IN7	Illuminazione 1 Liv +1		10	1
O.IN9	Illuminazione 1 Liv +1		14	1
O.IN11	Illuminazione 2 Sbarco Ascensori Liv +1		13	1
O.IN12	Illuminazione 2 Hall Liv +1		11	1
P15.IR1	Illuminazione 3 Sbarco Ascensori Liv +1		9	1
P15.IR3	Illuminazione 3 Hall Liv +1		12	1
P15.IR7	Illuminazione 1 Liv +1		6	1
P15.IR8	Illuminazione 2 Liv +1		9	1
S0.IS1	Illuminazione 1 Sicurezza	720 VA	12	1
S0.IS2	Illuminazione 2 Sicurezza	1440 VA	24	1
S0.IS3	Illuminazione Sicurezza	240 VA	4	1
S0.IS4	Illuminazione Sicurezza	660 VA	11	1

Figura 36: Estratto dell'abaco dei circuiti elettrici

I campi utilizzati per creare l'abaco sono i seguenti:

- Quadro
- Note circuito abaco
- Commenti
- Illuminazione - Unità abitativa collegati
- Numero di elementi
- Conteggio

L'abaco è stato successivamente riorganizzato per raggruppare i vari circuiti sotto ai relativi quadri elettrici. Ciò permette una visualizzazione più chiara e immediata.

Sono stati inseriti "Note circuito abaco" e "Commenti" per far apparire il codice assegnato al circuito e la relativa destinazione d'uso del locale servito in quanto, come già spiegato precedentemente, il programma assegna arbitrariamente un numero progressivo crescente al circuito che non rende facile e intuitiva la lettura.

È stata inserita la voce “Conteggio” per poter visualizzare quanti circuiti di quel tipo fossero collegati al quadro. Ciò permette di individuare agevolmente se vi sono più circuiti con lo stesso nome collegati al medesimo quadro.

L’intero abaco dei circuiti elettrici è presente all’allegato 12.4.

## 5.2.4 Confronto della modellazione delle luci con la normativa

Per quanto riguarda la modellazione delle luci, è stata data più attenzione all’aspetto delle informazioni piuttosto che quello geometrico. Sono state usate famiglie caricabili di default presenti all’interno delle librerie di Revit, e sono state assegnate solamente le dimensioni e la posizione corretta. Non è stata fatta una modellazione di dettaglio in quanto, a differenza di quanto fatto per lo strutturale, non si disponeva di sufficienti informazioni che potessero permettere ciò. È stata però data particolare attenzione all’aspetto delle informazioni.

Come si vedrà nei capitoli successivi, sono stati creati parametri condivisi tali da poter gestire le informazioni che verranno prese direttamente in loco tramite un’applicazione creata ad hoc.

prospetto 03a - Esempio di LOD dispositivo di illuminazione

LOD A	LOD B	LOD C	LOD D	LOD E	LOD F	LOD G
						
Geometria	Geometria	Geometria	<b>Geometria</b> Forma, dimensioni, ingombri ed inserimento di componenti di illuminazione effettivi. Effettivi margini ed ingombri per manutenzione, supporti, ancoraggi, per controllo vibrazioni e consolidamento antisismico.	<b>Geometria</b> Componenti supplementari per la fabbricazione e l’installazione in cantiere.	<b>Geometria</b> Come LOD E (rilevo di quanto eseguito).	<b>Geometria</b> Nuovi interventi: Come LOD F (con aggiornamenti) Manutenzione e gestione su elementi esistenti: Come LOD C o D (a partire da).
Oggetto	Oggetto	Oggetto	<b>Oggetto</b> Solido 3D	<b>Oggetto</b> Solido 3D	<b>Oggetto</b> Solido 3D	<b>Oggetto</b> Solido 3D
Caratteristiche	Caratteristiche	<b>Caratteristiche</b> - Definizione delle quantità su data sheet dei vari ambienti(piani)	<b>Caratteristiche</b> - Definizione effettiva di parametri di performance (lux, curve illuminotecniche, ecc.)	<b>Caratteristiche</b> - Nome prodotti, nome produttori - Modalità di installazione	<b>Caratteristiche</b> - Certificazione prodotto	<b>Caratteristiche</b> - Data di manutenzione/sostituzione - Soggetto manutentore - Storico delle manutenzioni

Figura 37: Descrizione LOD dei corpi illuminanti (UNI 11337-4:2017)

		Geometria	Oggetto	Restituzione Grafica
LOG	LOG D- Alto/Medi/Basso	Elemento tridimensionale. Dimensioni e forma effettivi. Collocato nella corretta posizione, con giusta quota e orientamento. Mancano elementi quali supporti e ancoraggi, questo è stato il motivo per cui il LOG è deificato come "D-".	Solido 3D semplificato	 <p>Vista 3D</p> <p>Vista in pianta</p> 

		Caratteristica elemento	Descrizione
LOI	Creazione	Caratteristiche circuito	Per ogni elemento è indicato il circuito a cui appartiene, riportando la medesima nomenclatura utilizzata nel progetto fornito, e il quadro a cui il circuito è collegato
		Caratteristiche Elemento	Per ogni elemento sono indicate le principali caratteristiche, come ad esempio il voltaggio, il tipo di luce emessa, il colore ecc..
	Manutenzione	Dati manutenzione	Ad ogni elemento sono collegati i parametri condivisi che identificano il tipo di manutenzione. Questi parametri condivisi verranno automaticamente compilati tramite il DB Link come si vedrà in seguito



## 6 LA REALTÀ AUMENTATA

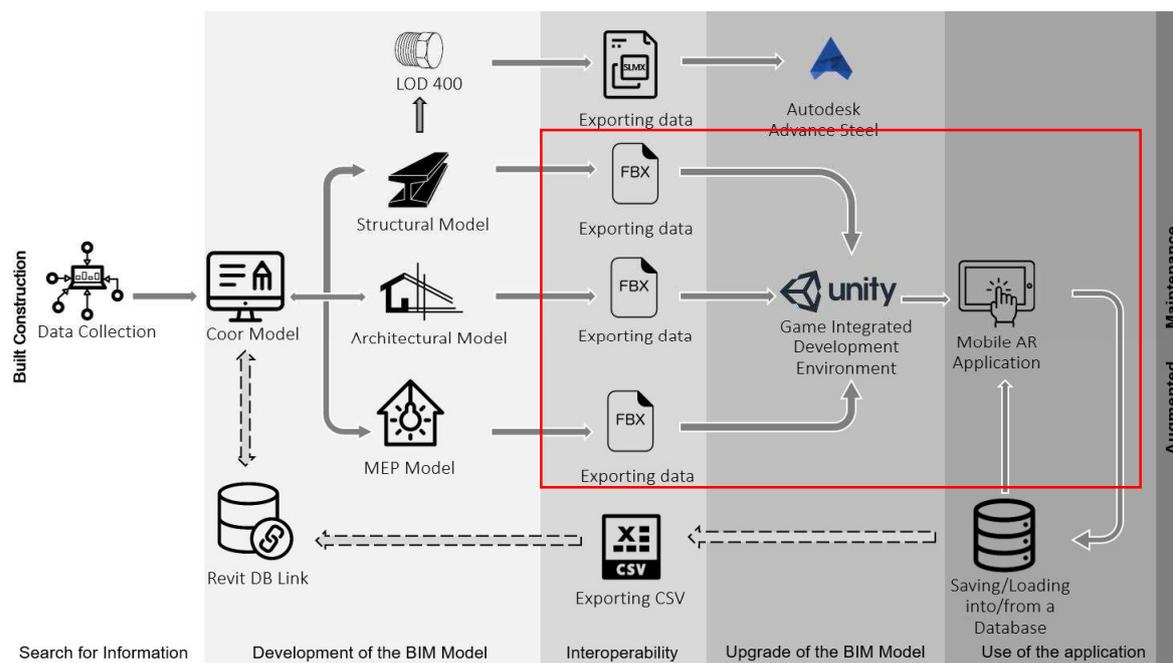


Figura 38: Posizione all'interno del workflow - Esportazione del modello e creazione dell'applicazione

### 6.1 Cos'è la Realtà Aumentata

Esistono 2 modi per visualizzare la realtà:

- La Realtà Virtuale
- La Realtà Aumentata

La prima consiste nella visualizzazione di un modello tridimensionale o di informazioni tramite un visore applicato direttamente di fronte gli occhi dell'utente. Inoltre alcuni visori supportano l'utilizzo di joystick wireless che l'utente, impugnandoli, può utilizzare per interagire con il modello.

La Realtà Aumentata invece consiste nella visualizzazione di informazioni su uno schermo (Tablet, Smartphone o computer) sovrapposte alle immagini ottenute dalla telecamera del dispositivo che inquadra la realtà. Queste due tecnologie, seppur diverse, hanno in comune lo scopo di far percepire all'utente la realtà che ci circonda in maniera diversa.

La differenza sostanziale tra le due è che la Realtà Virtuale proietta l'utente all'interno del modello visualizzato facendogli vivere un'esperienza immersiva. La Realtà Virtuale invece permette all'utente di inquadrare la realtà tramite un dispositivo mobile o un computer, e visualizzare informazioni sullo schermo relative a ciò che si sta inquadrando in quel momento.

### 6.1.1 Storia della Realtà Aumentata

Il primo dispositivo che utilizza la realtà aumentata appare nel 1968 ad opera di Ivan Sutherland. Egli produce i primi occhiali che sfruttano questa tecnologia. Negli anni '90 appaiono sul mercato nuovi dispositivi che integrano anche la geolocalizzazione, ma i primi veri prodotti utilizzabili dalla massa appaiono negli anni 2000.

Già usata in ambiti molto specifici come militare, medico o nella ricerca, nel 2009 grazie al miglioramento della tecnologia la realtà aumentata è arrivata al grande pubblico sia come campagne di comunicazione augmented advertising pubblicate sui giornali o sulla rete, sia attraverso un numero sempre crescente di applicazioni per telefonini, in particolare per Windows Phone, Android e iPhone.

È oggi infatti possibile con la realtà aumentata trovare informazioni rispetto al luogo in cui ci si trova (come alberghi, bar, ristoranti, stazioni della metro) ma anche visualizzare le foto dai social network o voci Wikipedia sovrapposte alla realtà; ritrovare la macchina parcheggiata; taggare luoghi, inserire dei messaggi in realtà aumentata in un luogo specifico<sup>2</sup>.

### 6.1.2 Tipi di Realtà Aumentata

Esistono due tipi principali di realtà aumentata:

- Realtà aumentata su dispositivo mobile. Il telefonino (o smartphone di ultima generazione) deve essere dotato necessariamente di Sistema di Posizionamento Globale (GPS), di magnetometro (bussola) e deve poter permettere la visualizzazione di un flusso video in tempo reale, oltre che di un collegamento Internet per ricevere i dati online. Il telefonino inquadra in tempo reale l'ambiente circostante; al mondo reale vengono sovrapposti i livelli di contenuto, dai dati da Punti di Interesse (POI) geolocalizzati agli elementi 3D.

- Realtà aumentata su computer. È basata sull'uso di marcatori, (ARtags), di disegni stilizzati in bianco e nero che vengono mostrati alla webcam, vengono riconosciuti dal computer, e ai quali vengono sovrapposti in tempo reale i contenuti multimediali: video, audio, oggetti 3D, ecc. Normalmente le applicazioni di realtà aumentata sono basate su tecnologia Adobe Flash.

Esistono inoltre diversi tipi di tracciamento. Il tracciamento dell'immagine è il processo che consente la comprensione dell'immagine osservata per una corretta combinazione di questa immagine con gli elementi virtuali corrispondenti.

---

<sup>2</sup> Fonte: Wikipedia – Realtà Aumentata

Questi tipi di follow-up sono andati evolvendosi di pari passo con la tecnologia e oggi è possibile identificare i seguenti tipi di follow-up:

- Sensori

L'uso di sensori è stata la prima forma di follow-up nello sviluppo di strumenti di realtà aumentata. Come suggerisce il nome, questo si basa sull'uso di sensori (giroscopi, GPS, Accelerometri, Wifi, Bluetooth, bussola, ecc.)

- Marker

I marker sono simboli stampati (codici QR) o immagini che possono essere rilevati dalla fotocamera e interpretati dal software per la sovrapposizione di elementi virtuali su di essi.

La comprensione dell'ambiente è determinata dalla posizione in cui viene rilevato il marcatore. Ciò implica che nella maggior parte dei casi gli elementi virtuali scompaiono quando l'immagine del marker scompare dall'inquadratura.



Figura 39: Esempio di un edificio visualizzato in AR con il metodo dei marker – Fonte: [www.augmentend.com](http://www.augmentend.com)

- Marker per oggetti 3D

A differenza dei marker convenzionali, in questo caso vengono utilizzati oggetti fisici tridimensionali. In assenza dell'uso di un'immagine specifica questo tipo di tracciamento richiede una maggiore capacità di calcolo e quindi dispositivi migliori. Per la sua corretta operazione è necessario che l'oggetto sia opaco, rigido e senza parti mobili.



*Figura 41: Marker tridimensionale – Fonte : Video Youtube di Parth Anand - Augmented Reality Shopping.*



*Figura 41: Apple Watch visualizzato in AR tramite marker tridimensionale*

- Marker per ambiente

Conosciuta anche come realtà aumentata senza marker, funziona usando l'informazione visiva dell'ambiente che individua da sé i modelli nell'ambiente e li interpreta per raggiungere una comprensione della sua posizione. Questo include riconoscimento facciale, oggetti e superfici senza l'uso di marcatori pre-programmati. Rappresenta il tipo di tracciamento più efficiente attualmente presente grazie alla sua versatilità.



Figura 42: Esempio di marker per ambiente – Fonte: [www.apple.com/arkit/](http://www.apple.com/arkit/)

- Ibridi

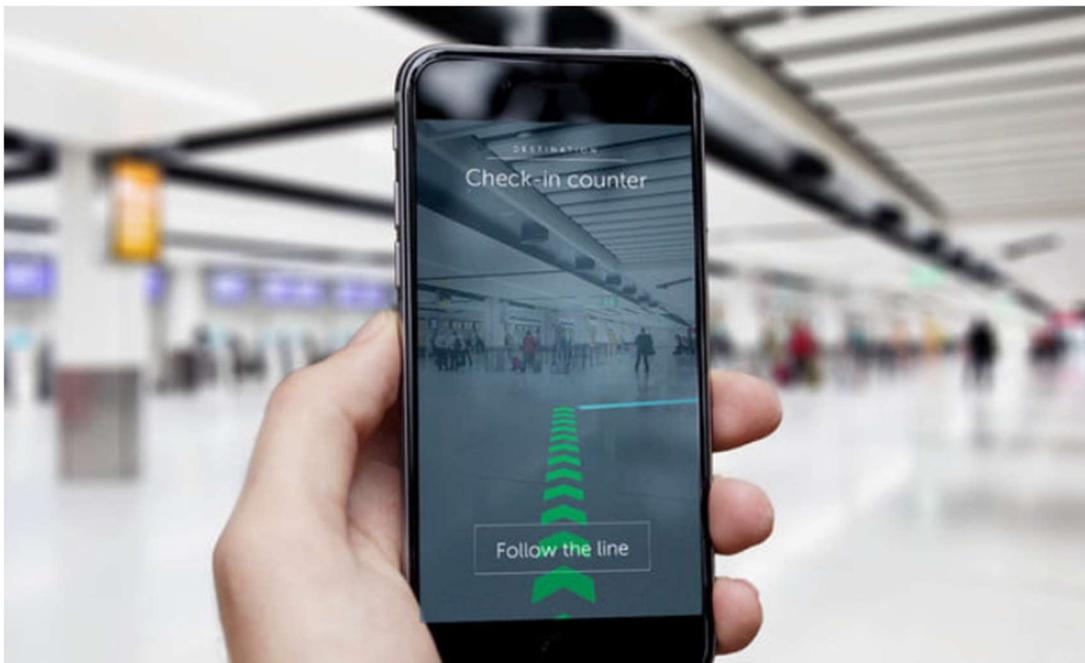


Figura 43: Esempio di marker ibrido – Fonte: <http://techtrends.tech/tech-trends/gatwick-beacons/>

Il tracciamento ibrido utilizza sensori per ottenere dati di posizione e orientamento e integra anche tecniche di tracciamento visivo. Sono generalmente utilizzati per attivare il modello che deve essere visualizzato quando si arriva in una posizione specifica e quindi avviare il tracciamento per mezzo di alcune delle tecniche di cui sopra.<sup>3</sup>

<sup>3</sup> Fonte: La realidad aumentada como medio de visualización del modelo BIM en la construcción – Gerargo Chavarri

## 6.2 Applicazione al caso studio

Quanto visto fino ad ora in linea teorica è stato successivamente applicato ad un caso reale, nella fattispecie al centro servizi della torre Regione Piemonte visto nei capitoli precedenti. Dato che l'applicazione è pensata per dispositivi mobili si è scelto di trascurare molte informazioni appartenenti al modello in quanto sia su iPhone7 che su iPad Pro 9.7" si dispone solo di 2 Gb di RAM.

### 6.2.1 Esportazione dei modelli Revit in fbx

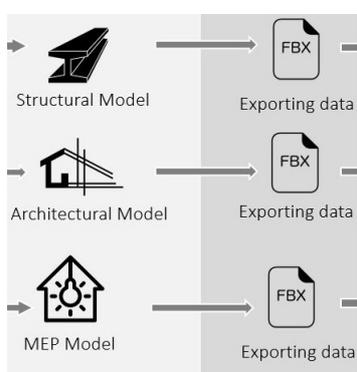


Figura 44: Posizione all'interno del workflow - Esportazione

Per prima cosa è stato esportato il modello da Revit in formato “fbx”.

“FBX (Filmbox) è un formato file proprietario (.fbx) sviluppato da Kaydara e di proprietà di Autodesk dal 2006. Viene utilizzato per fornire l'interoperabilità tra le applicazioni di creazione di contenuti digitali. FBX è anche parte di Autodesk Gameware, una serie di video game middleware.

La FBX può essere rappresentato su disco sia come dati binari sia come ASCII; il suo SDK supporta la lettura e la scrittura di entrambi.

Per il formato di file binario FBX, Blender Foundation ha pubblicato una specifica ufficiale, così come una specifica di livello superiore non ufficiale (lavori in corso) per come i dati reali sono disposti in FBX (indipendentemente dal formato ASCII o binario). “[3]

L'esportazione dei file è avvenuta in più fasi in quanto si è diviso il modello per discipline:

- Architettonico
- Strutturale
- MEP Antincendio
- MEP Elettrico

[3] Fonte: Wikipedia - FBX

Del modello strutturale non sono stati esportati i giunti in acciaio in quanto successivamente si è visto che questi gravavano troppo sulla RAM del dispositivo mobile portandolo a crashare in fase di apertura dell'applicazione.

Ciò però non è risultato essere un problema dato che l'applicazione è pensata per la manutenzione degli impianti e quindi perdere informazioni sulle strutture non ha comportato nessuno discostamento dagli obiettivi preposti.

Una volta esportati i modelli da Revit sono stati importati in Unity3D.

## 6.2.2 Test 1: Utilizzo della Realtà Aumentata con Marker

Come primo approccio alla realtà aumentata è stato fatto un test con Vuforia, per vedere come quest'applicazione funzionasse e quanto effettivamente potesse essere utile per perseguire l'obiettivo prefissato. Dato che, appunto come detto prima, si trattava di un test, si è importato solamente il modello dello strutturale per una maggiore velocità e leggerezza dell'applicazione.

Per importare un file bisogna cliccare su Assets > Import new Assets.

Successivamente bisogna importare la telecamera per la realtà aumentata.

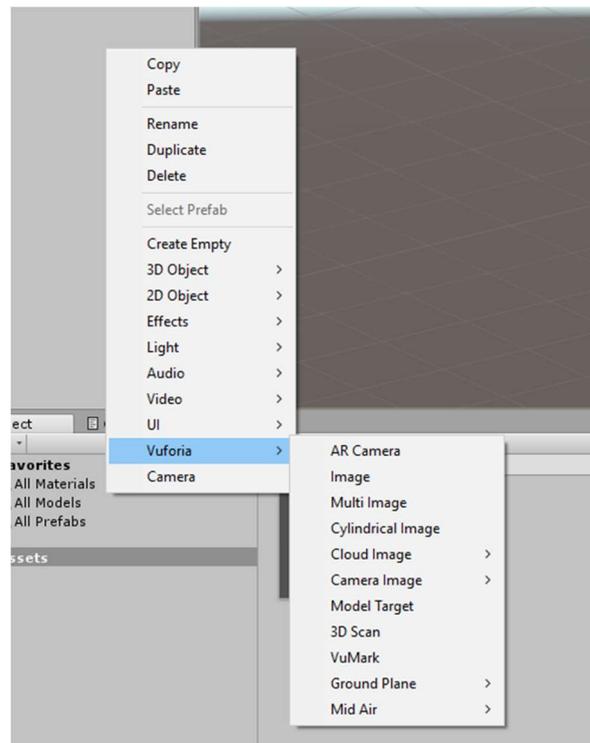


Figura 45: Importazione AR Camera

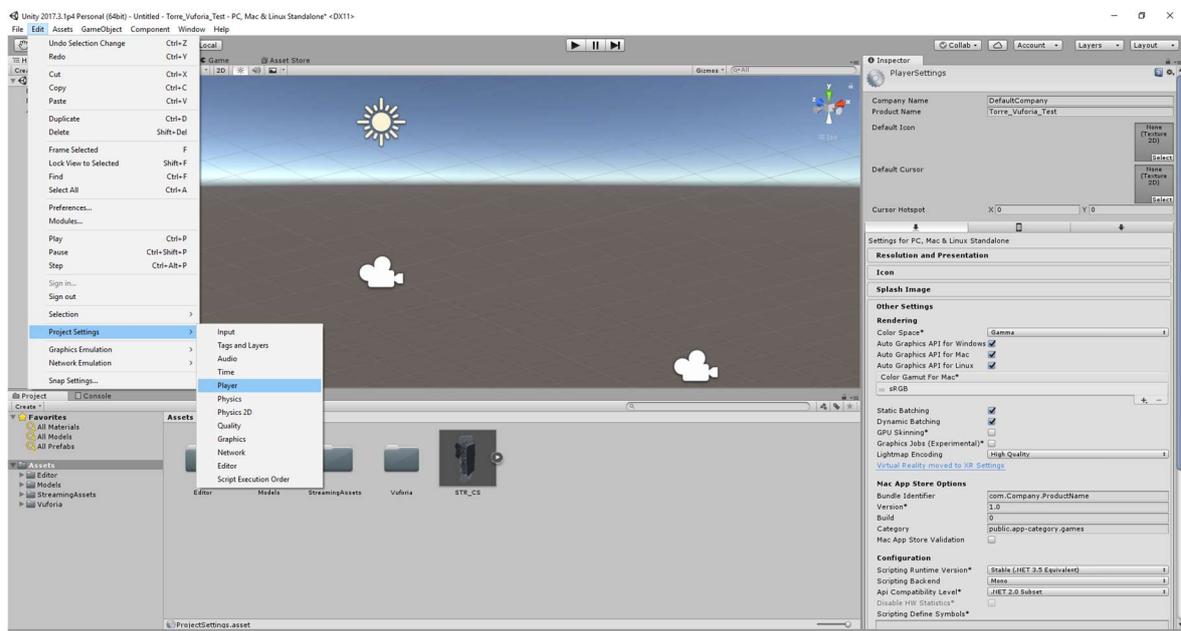


Figura 46: Impostazioni Player Settings in Unity3D

Se si utilizzasse una versione precedente a Unity 2017 bisognerebbe invece importare un nuovo Assets dall'assets store. Successivamente sono state modificate alcune impostazioni del "Player Settings".

Per accedere a queste proprietà bisogna cliccare su: Edit > Player Settings > Player.

Nell'Inspector è stato modificato il "Bundle Identifier" con un nome univoco per l'applicazione.

Il "Device Filter" è stato cambiato in "ARMv7" ed è stata tolta la spunta affianco ad "Android TV Compatible".

Successivamente si è passati ad inserire la licenza dell'applicazione.

Per fare ciò si è aperto innanzitutto il "Vuforia Configuration" tramite l'om-

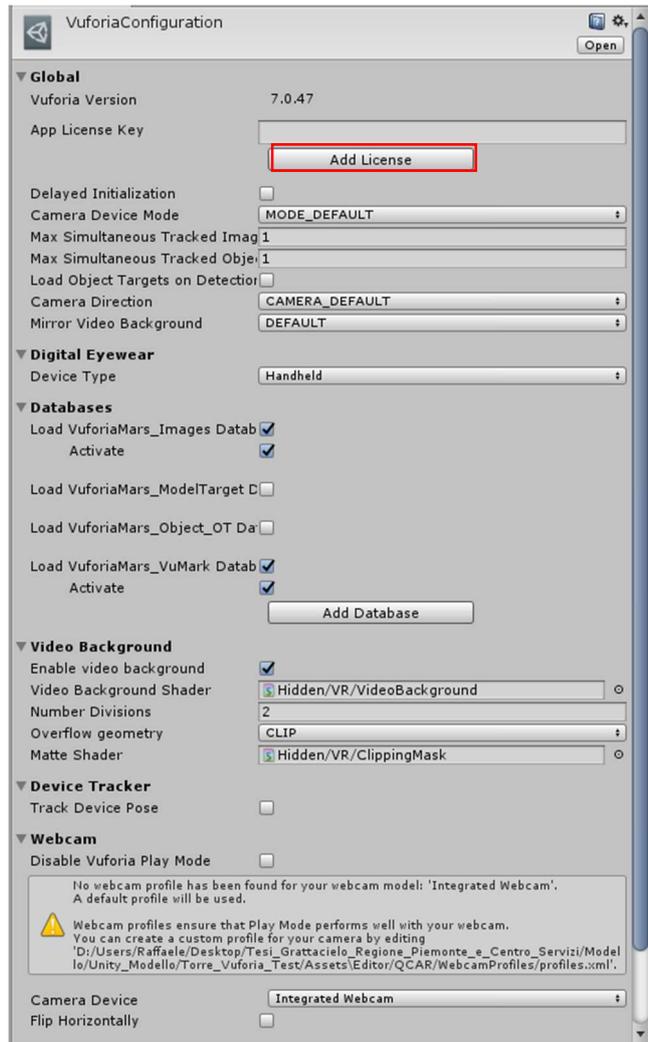


Figura 47: Vuforia Configurator

nimo bottone presente nella schermata precedente. La licenza la si può recuperare accedendo al sito web [www.vuforia.com](http://www.vuforia.com). Una volta entrati nel sito bisogna accedere alla sezione per sviluppatori denominata "Dev Portal" e successivamente in "Develop".

Una volta fatta la registrazione o il log-in, si troverà una schermata con due collegamenti denominati “License Manager” e “Target Manager”.

Nella prima si troverà la licenza mentre nella seconda il pacchetto da scaricare per impostare l’image target.

Per creare una licenza bisogna prima creare un nuovo progetto cliccando su “Get Development Key”.

## Torre Regione Piemonte [Edit Name](#) [Delete License Key](#)

License Key
Usage

Please copy the license key below into your app

```

AUYOjjP/////AAAAMX2GiyaQrEg9h0dMB7RGJCIANB0J7QuC+3yS
lcaIhZZd2s5EkLZ8WunP5+0oRmnOm/MxGAX9kRikoqMLHQiRwdiR
vylCU2nGRWzrDZmWzcIvZ+IuyFxnPv4NL8+CwWFxfliJoroakaob
pDTquRSQvKCLDEL2TShsgHQ4yvoaV5sPOwxrLGsqj6mZOebD+IzF
GTK+Doz1TCO+rD4aTUysOfuGo83CNX5y/1KsWmy0Plkj+eoGIVcW
m+MEvnZNzXOA4KexPxbKEjG4xXt9wnuMeXFGHE6W+yBAQL7w1+Er
GQ1Z7HqcXhGV8tRoyxA4tIS0SL/ZVNmyTnEcBUEYy+wPT1Jw2Rn1
v8fiSb//J5Q+pCL1

```

**Type:** Develop  
**Status:** Active  
**Created:** Mar 24, 2018 16:17

**History:**  
License Created - Mar 24, 2018 16:17

Figura 48: Licenza sul sito di Vuforia

License Manager
Target Manager

Target Manager > Torre\_Regione\_Pie...

## Torre\_Regione\_Piemonte [Edit Name](#)

**Type:** Device

Targets (1)

Add Target

Download Database (All)

<input type="checkbox"/> Target Name	Type	Rating	Status	Date Modified
<input type="checkbox"/> CS_L_0	Single Image	★ ★ ★ ★ ★	Active	Mar 24, 2018 16:59

Figura 49: Creazione Image Target

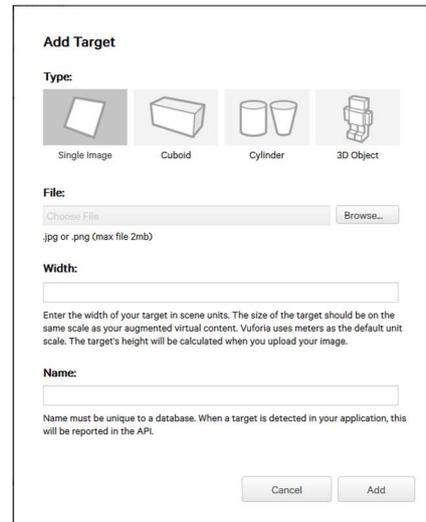
Fatto ciò, sono state create le immagini target.

In questo caso ne è stata creata solo una perché, come già detto, si trattava solo di un test.

Per creare l'immagine target bisogna cliccare su "Target Image" e, in questa nuova schermata, selezionare la voce "Add Database". Nella schermata che apparirà bisogna scegliere un nome, spuntare la voce "VuMark" e scegliere l'applicazione a cui collegare questo database; in questo caso è stato scelto il progetto creato in precedenza.

Aperto il "Target Manager" relativo al progetto corrente, cliccando su "add" è possibile finalmente caricare l'immagine di riferimento. L'applicazione consente di inserire sia immagini, ma anche oggetti tridimensionali. In questo caso si è scelto di inserire come immagine di riferimento la pianta dell'architettonico

livello 0 del centro servizi. Una volta caricata l'immagine il sito farà dei processi per determinare i punti noti dell'immagine a cui verrà collegato il modello. È ora possibile scaricare



**Add Target**

Type:

Single Image   Cuboid   Cylinder   3D Object

File:

Chosen File   Browse...

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

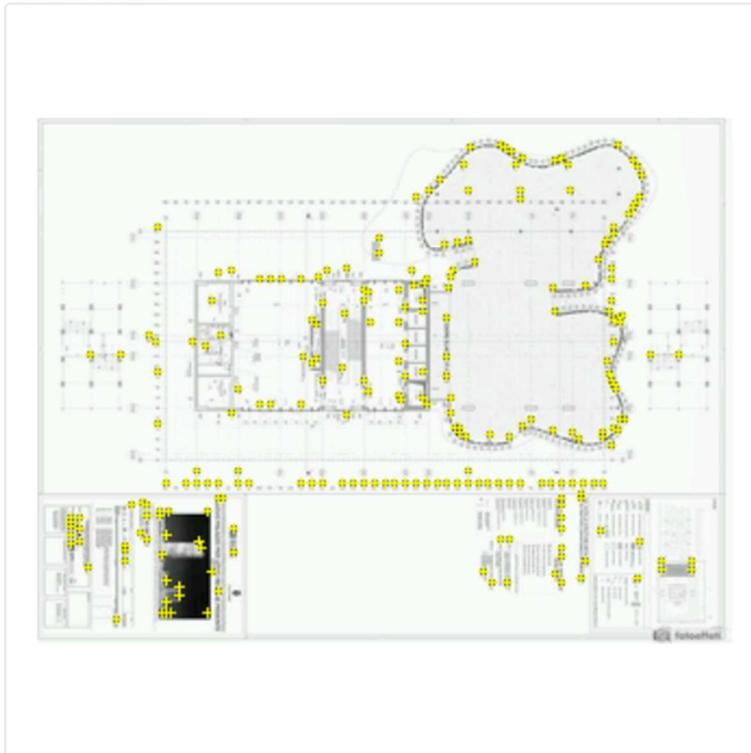
Cancel   Add

Figura 50: Add Target

Target Manager > Torre\_Regione\_Pie... > CS\_L\_0

## CS\_L\_0

Edit Name   Remove



Type: Single Image

Status: Active

Target ID: 90590795caf34b418ead6de0f31a4917

Augmentable: ★★☆☆☆

Added: Mar 24, 2018 16:59

Modified: Mar 24, 2018 16:59

Update Target   Hide Features

Figura 51: Punti noti dell'Image Target

il pacchetto cliccando sul link del progetto e cliccando su “download Database (All)”. È possibile vedere i punti noti individuati dall’applicazione per decidere se sono sufficienti o meno cliccando sull’immagine e cliccando su “Show Features”.

Scaricato il pacchetto, bisogna importarlo in Unity3D tramite Assets > Import Package > Custom Package.

Una volta importato il pacchetto, cliccando col tasto destro nella sezione “Hierarchy” e selezionando Vuforia > Image si è importata l’Image Target di default.

Per modificare quest’immagine bisogna aprire l’Inspector dell’Image Target e nella

sezione Image Target Default > Database si seleziona l’immagine corretta. Una volta fatto ciò con un semplice drag and drop si trascina il modello precedentemente caricato del centro servizi dalla sezione Assets in quella denominata “Scene”. Una volta fatto ciò basterà spostare e scalare opportunamente la foto per allinearla al meglio con il modello facendo coincidere, ad esempio, i pilastri del modello con i pilastri rappresentati in tavola. Il risultato risulta essere come quanto rappresentato in figura 42.

Per verificare il corretto funzionamento dell’applicazione basta cliccare sul tasto “Play” posto nella parte superiore della schermata “Scene”.

Unity3D avvierà l’applicazione sfruttando la webcam del computer. Dato che si trattava di un semplice test per verificare se l’applicazione funzionasse o meno, non si è installata quest’ultima su un dispositivo mobile, ma si è fatto il debug direttamente da Unity3D. Come è possibile notare dal video, non appena l’immagine target sparisce dall’inquadratura, totalmente o in parte, il modello scompare e, per farlo riapparire, è necessario inquadrare nuovamente l’immagine target e aspettare che il modello venga ricaricato e visualizzato.

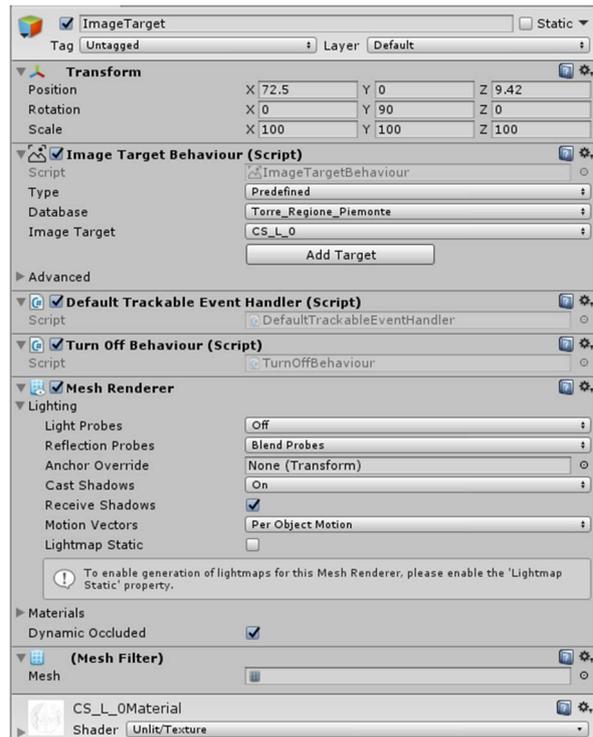
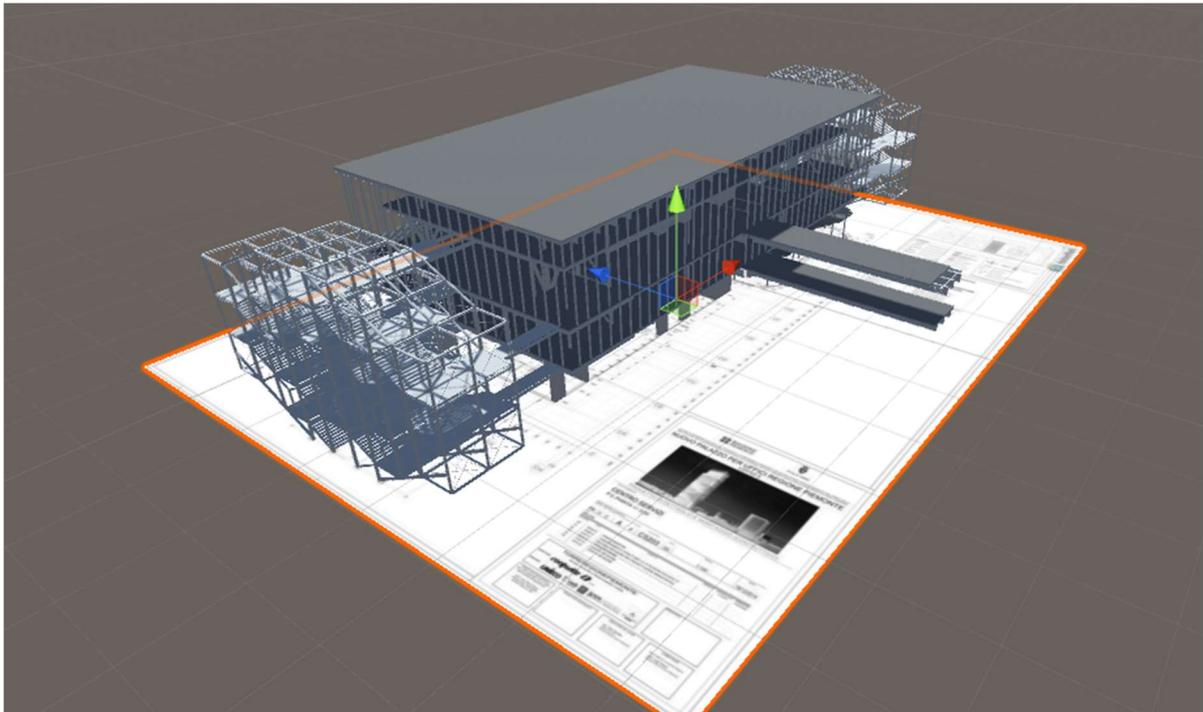


Figura 52: Inspector dell’Image Target



*Figura 53: Risultato Allineamento Image Target e modello strutturale*

Questa soluzione non è adatta allo scopo prefissato in quanto, se si volesse esplorare l'edificio, non si avrebbe sempre la possibilità di inquadrare un'immagine target e quindi il modello perderebbe costantemente il riferimento e scomparirebbe dallo schermo. Quest'applicazione comunque risulta avere utilità, ad esempio, per dare più informazioni alle tavole da disegno. Infatti si potrebbe pensare, in futuro, di allegare alle tavole dei video esplicativi su come eseguire particolari operazioni o inserire documenti relativi alle proprietà degli elementi raffigurati.

Accedendo al video collegato al codice QR riportato è possibile osservare il modello strutturale del centro servizi legato tramite marker alla tavola del primo piano fuori terra del centro servizi. [Figura 54]

Per raggiungere invece l'obiettivo prefissato si è deciso di utilizzare un altro plug-in di Unity3D: Apple ARKit.



*Figura 54 QR Code test Vuforia*

## 6.2.3 Test 2: Utilizzo della Realtà Aumentata con Marker per ambiente

Apple ARKit è un plug-in sviluppato da Apple che sfrutta la realtà aumentata di tipo “Marker per ambiente”. Questa applicazione è disponibile solo per dispositivi dotati di sistema operativo iOS 11.3 o successivi. È possibile utilizzare l’ARKit sia su Unity3D che su Xcode; in questo



*Figura 55: Logo Apple ARKit*

caso è stato utilizzato su Unity3D in quanto su questa piattaforma il procedimento di programmazione in C# è più snello e richiede competenze più basilari rispetto a Xcode. Sono stati fatti diversi test con questa applicazione in quanto, essendo un’applicazione rilasciata a settembre 2017, mentre si lavorava allo sviluppo dell’applicazione stessa, venivano rilasciati costantemente aggiornamenti che risolvevano vari bug o miglioravano le performance stesse dell’applicazione.

### 6.2.3.1 Importazione dell'ARKit e dei modelli FBX

Creato il progetto di Unity3D denominato “Torre Regione Piemonte”, per prima cosa è stato importato l'asset dell'ARKit direttamente dall'Asset Store presente in unity3D.

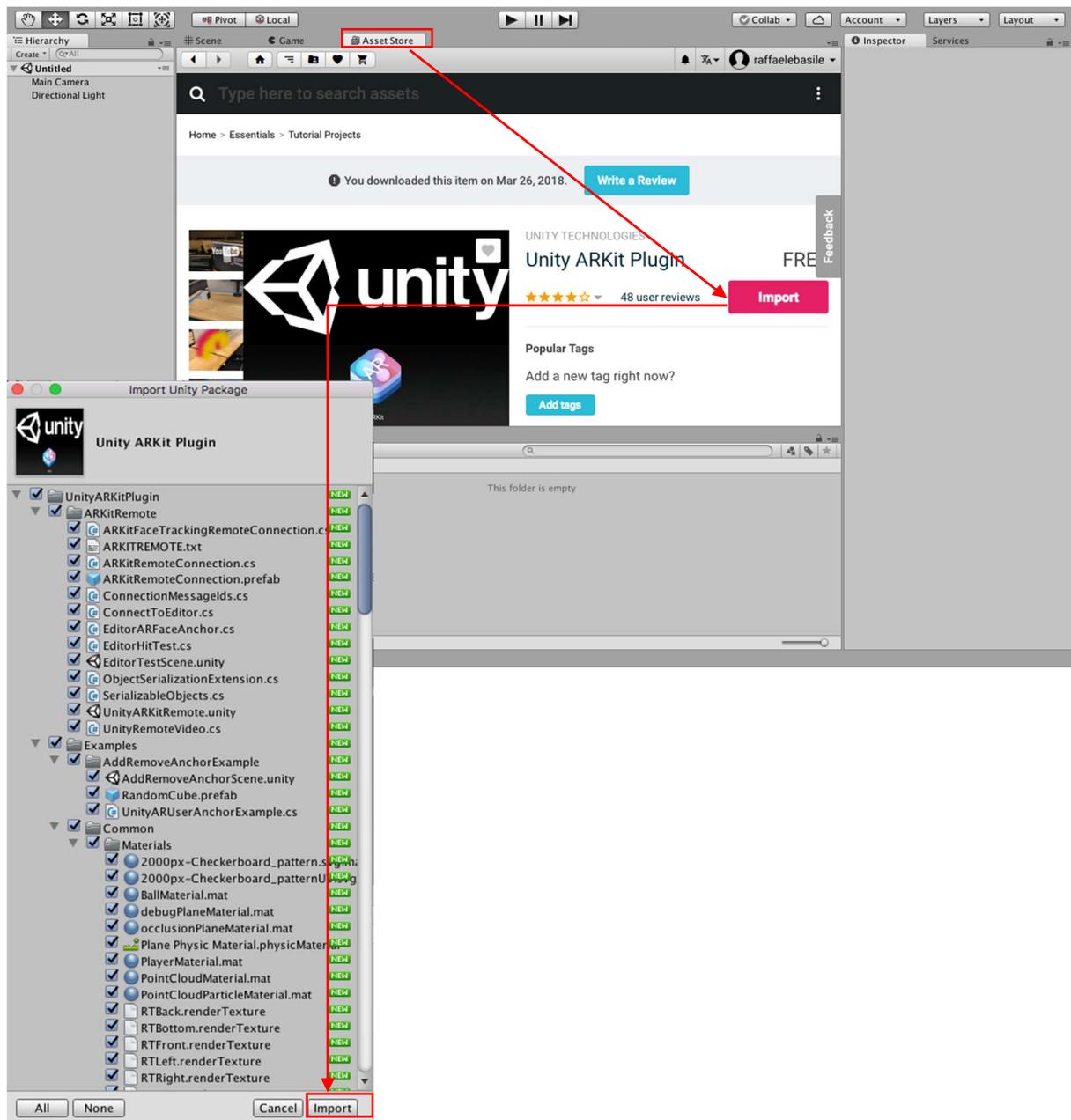


Figura 56: Importazione Apple ARKit

Per fare ciò basta cliccare su Asset Store, cercare l'ARKit e importare tutti gli elementi presenti nel pacchetto. Una volta fatto ciò, tornando nella schermata “Scene” è stata creata una cartella denominata “Modello” all'interno del quale sono stati inseriti i 4 file FBX relativi allo strutturale, architettonico, elettrico e antincendio.

Per importare i file all'interno del progetto è stato fatto un "drag and drop" analogamente a quanto fatto nel caso di Vuforia.

Installando l'applicazione, come si vedrà nel paragrafo successivo, si è notato che l'applicazione crashava all'avvio in quanto il modello risulta essere molto pesante e, né l'iPhone né l'iPad, dispongono di una RAM sufficiente per eseguire l'applicazione. Per ovviare a questo problema sono state compresse le Mesh. Per fare ciò basta cliccare sui singoli modelli e, nell'Inspector, andare a selezionare Mesh Compression.

Per il file architettonico è stata scelta una compressione media, mentre per il modello strutturale è stata scelta una compressione alta in quanto risultava essere il modello più pesante. La compressione delle mesh comporta una perdita di informazioni a livello geometrico in quanto gli elementi vengono semplificati per pesare di meno. Ciò non è risultato essere un problema in quanto le informazioni necessarie sono solo sugli impianti; infatti per quest'ultimi non è stata applicata nessuna compressione.

Inoltre, al fine di risparmiare ulteriore RAM, sono state tolte tutte le texture al file architettonico e strutturale, mentre per quanto riguarda gli impianti, sono stati creati due materiali, denominati Antincendio di colore rosso e Elettrico di colore giallo, in modo tale da identificarli più semplicemente durante l'esecuzione dell'applicazione.

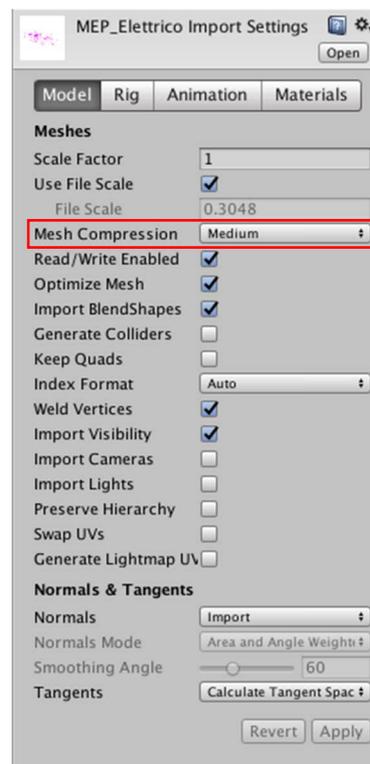


Figura 57: Mesh Compression del file architettonico

Innanzitutto è stata creata una cartella denominata “Materiali” e all’interno sono stati creati i due materiali. Successivamente andando a selezionare il file degli impianti, nella sezione denominata “Materials” è stato selezionato il relativo materiale.

Una volta impostati tutti i materiali e importati i pacchetti necessari è stato possibile passare alla creazione dell’applicazione.

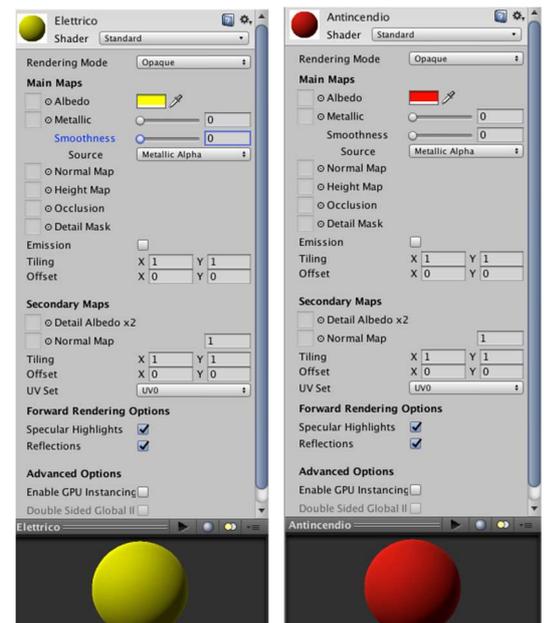


Figura 58: Inspector dei materiali

### 6.2.3.2 Impostazione della scena per la realtà aumentata

Una volta importati i pacchetti e i modelli e impostati tutti i vari parametri, è stata aperta la “scena” adatta a creare l’applicazione per la realtà aumentata.

Per aprire la scena bisogna andare in Assets >

UnityARKitPlugin > Examples > UnityARKitScene.

Aperta questa scena l’elenco degli elementi presenti nella gerarchia cambierà. Si può notare innanzitutto come la “Main Camera” presente in precedenza ora si trovi all’interno del “CameraParent”.

Questa è la telecamera che sfrutta la realtà aumentata. Inoltre si può notare la presenza di un nuovo elemento nella gerarchia: l’“HitCubeParent”.

All’interno di questo elemento andranno inseriti i modelli da visualizzare in realtà aumentata. Di default vi è un “HitCube”.

Di per sé questo elemento non ha alcuna utilità ai fini del modello e, volendo, lo si potrebbe eliminare, ma in questo caso è stato lasciato in quanto si è notato che l’applicazione setta l’origine del modello proprio nella posizione di questo elemento. Il passo successivo è stato quello di aggiungere i singoli modelli all’interno della scena.

Per fare ciò si selezionano i modelli interessati e si trascinano nella gerarchia all’interno del “HitCubeParent” come da figura.

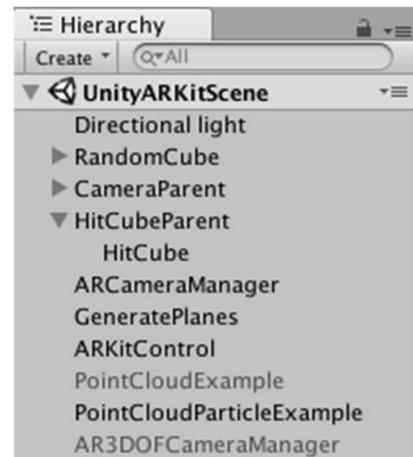


Figura 59: Hierarchy dell'ARKit

Una volta fatto ciò si è quasi pronti per installare l'applicazione su dispositivo mobile. L'ultimo passo da fare è quello di aggiungere uno script ai singoli modelli che permetta all'applicazione di collocare i singoli elementi nello spazio.

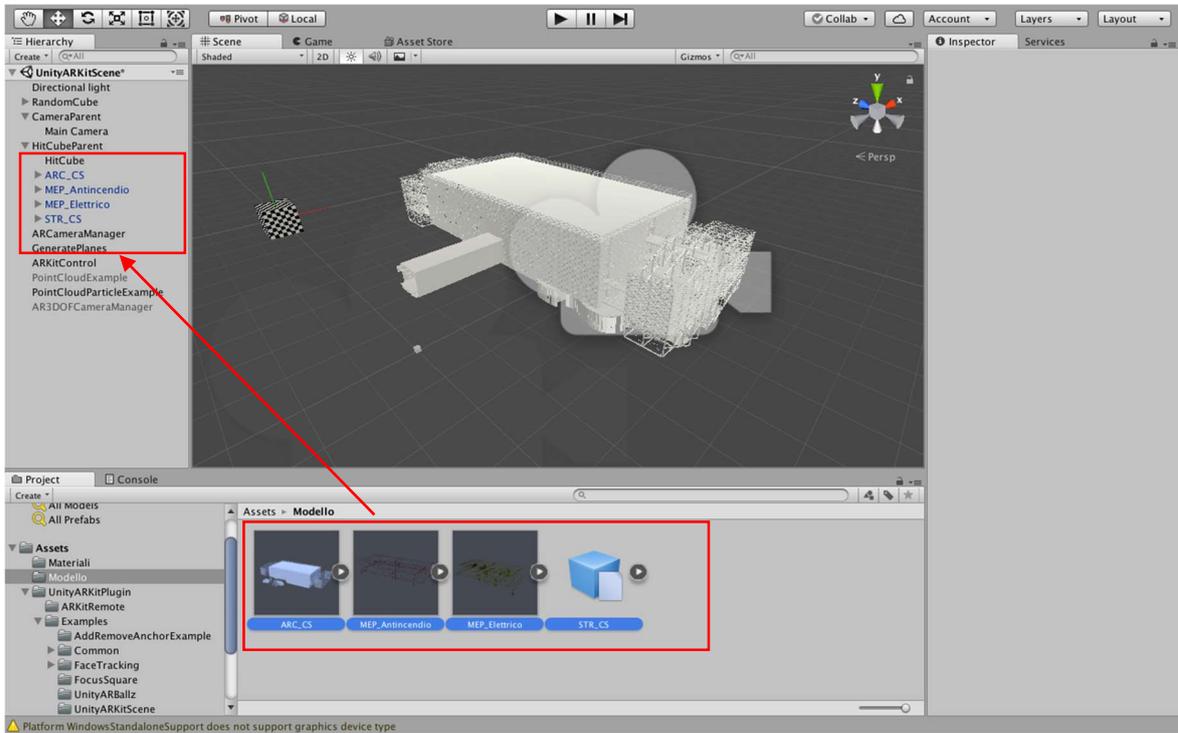


Figura 60: Modelli importati nella scena

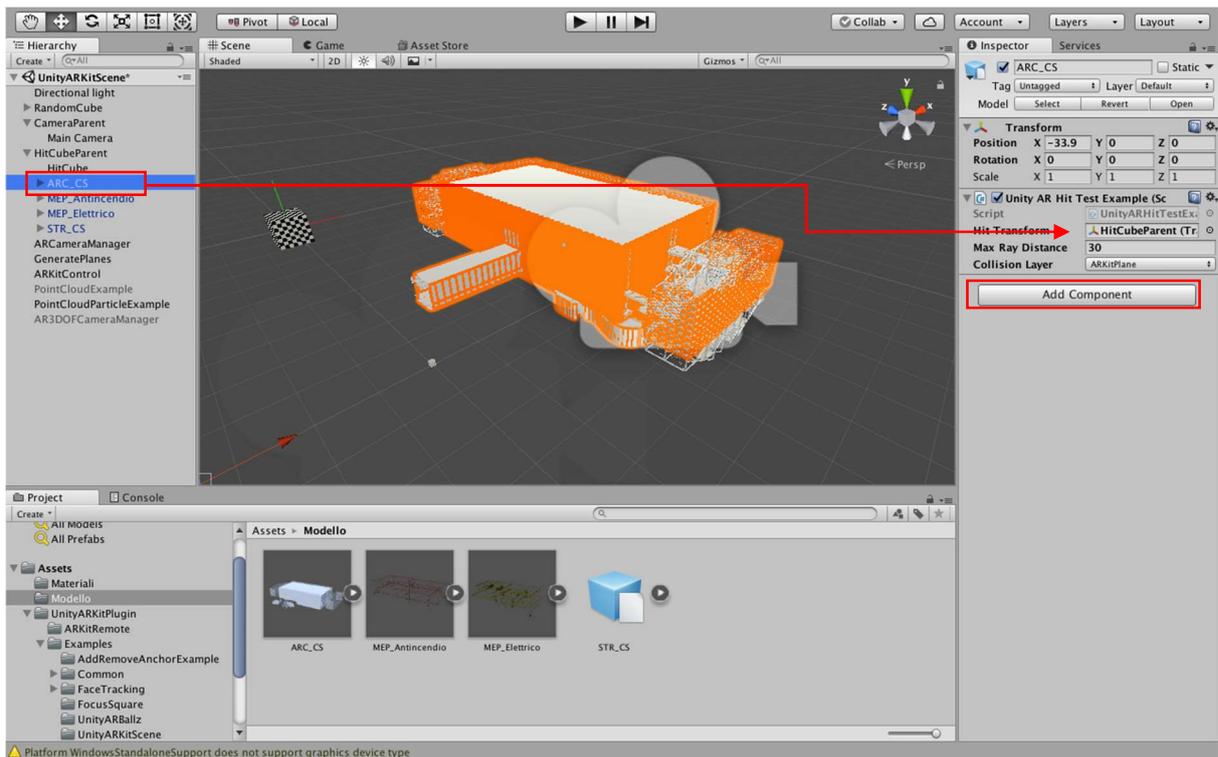


Figura 61:Unity AR Hit Test Example script

Per fare ciò basta selezionare un modello per volta dalla gerarchia e nell'Inspector selezionare "Add Component" e ricercare lo script chiamato "UnityARHitTestExample". Successivamente bisogna trascinare l'HitCubeParent all'interno del campo denominato "Hit Transform". Una volta fatto questo procedimento per tutti i modelli importati, si può procedere a installare l'applicazione su dispositivo mobile.

### 6.2.3.3 Installazione dell'applicazione su dispositivo mobile mediante Xcode

Per installare l'applicazione su iPhone o iPad bisogna esportare il progetto su Xcode.



Figura 62: Logo Xcode

Xcode è l'ambiente di programmazione Apple tramite il quale si programmano le applicazioni che si trovano sull'Apple Store, ma anche applicazioni private, come in questo caso, ovvero che non potranno essere scaricate da chiunque ma solo da chi ha i permessi necessari. Questo va a tutela del committente in quanto i dati contenuti all'interno dell'applicazione sono dati sensibili a cui non può avere accesso chiunque.

Xcode supporta numerosi linguaggi di programmazione tra cui, per citarne alcuni, Swift, C, C++, C# ecc...

In questo caso non ci si avvale di Xcode per programmare ma solo per installare l'applicazione sull'iPhone o sull'iPad.

Per fare ciò bisogna prima impostare dei settaggi in Unity3D.

Per prima cosa bisogna accedere al “Build Settings” Tramite la scheda “File”.

In questa schermata compaiono tutti i dispositivi sui quali è possibile andare ad installare l'applicazione. In questo caso però è possibile installare solo su dispositivi che dispongono di un sistema operativo iOS, in quanto la Apple, che è un'azienda molto chiusa nel proprio ecosistema, non permette ad altri sistemi operativi di utilizzare le proprie applicazioni. Per far fronte a ciò Google ha risposto all'ARKit con un'applicazione propria chiamata ARCore. Quest'ultima non è stata testata nel presente lavoro in quanto essendo un'applicazione di Google non può essere eseguita su dispositivi Apple.

Dunque, per prima cosa bisogna selezionare la piattaforma sulla quale si vuole andare ad installare l'applicazione, e successivamente cliccare su “Switch Platform”.

Questo processo impiegherà molto tempo. Una volta finite quest'operazione, cliccando su “Player Settings”, nella medesima schermata, si andrà a modificare il campo relativo al “Bundle Identifier”, inserendo una voce univoca, in questo caso è stato inserito “Raffaele-Basile.TorreRegionePiemonte.CentroServizi”, e in corrispondenza di “Target Minimum iOS Version” si andrà ad inserire 11.3 che risulta essere il numero della versione iOS meno recente per cui la realtà aumentata è supportata.

È possibile inoltre andare a selezionare il logo dell'applicazione nell'omonimo campo. In questo caso è stato scelto il logo presente nel cartiglio delle tavole.

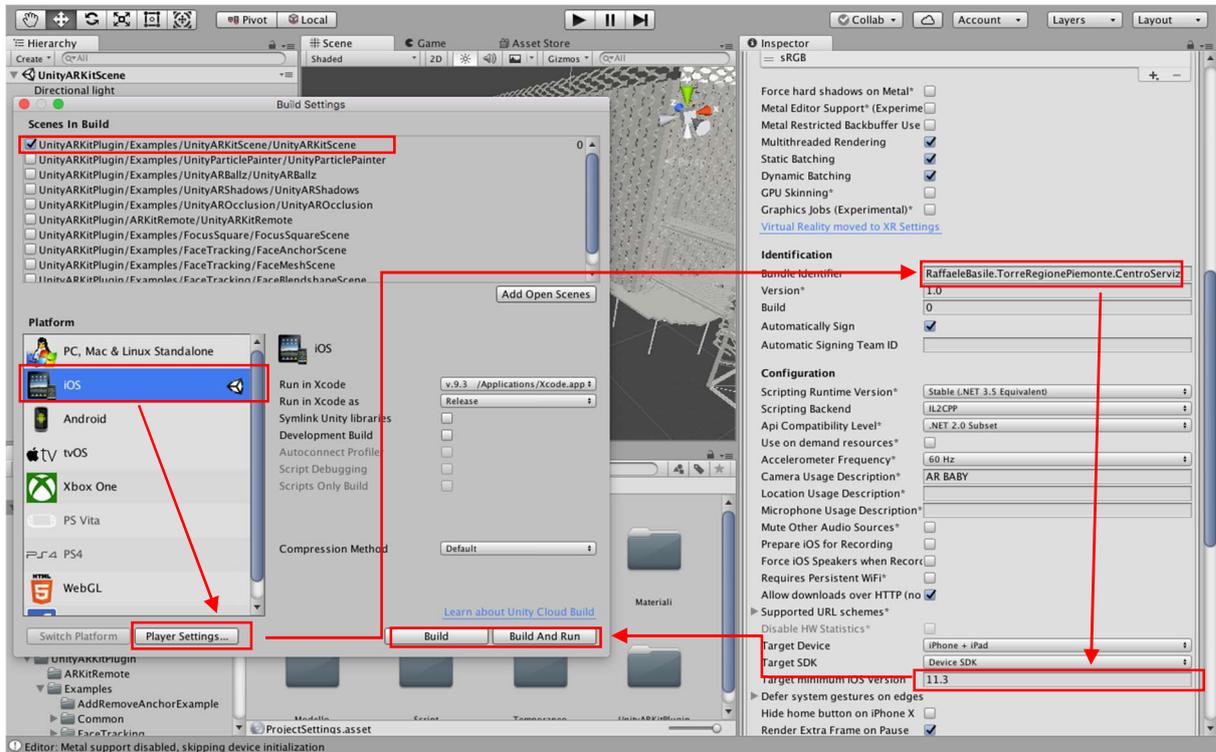


Figura 63: Impostazioni in Unity3D prima dell'esportazione in Xcode

Infine nel campo “Scenes In Build” bisogna andare a selezionare la scena che si desidera esportare. In questo caso è stata scelta quella denominata “UnityARKitScene”.

Fatto ciò basterà cliccare su “Build” e il processo di esportazione partirà. Una volta completato questo processo si aprirà in automatico Xcode.

In questo ambiente di programmazione non si andrà a scrivere alcun codice.

Innanzitutto si seleziona il dispositivo su cui installare l'applicazione (che necessariamente deve essere collegato al Mac con il cavo USB-Lightning) e in seguito il Team di sviluppo dell'applicazione.

In questo caso non essendoci un team di sviluppo vero e proprio si è usato l'ID Apple del sottoscritto. Nella Parte superiore dello schermo si può scegliere il nome che apparirà sotto l'icona dell'applicazione e la versione della stessa.

Nella parte inferiore invece si possono andare a settare alcuni parametri quali ad esempio la possibilità di eseguire l'applicazione in landscape o in Split-View.

È molto importante non andare a modificare la voce contenuta nel “Bundle Identifier” in quanto se questa non dovesse coincidere con la voce inserita in Unity3D, Xcode non riuscirebbe ad installare l’applicazione sul dispositivo.

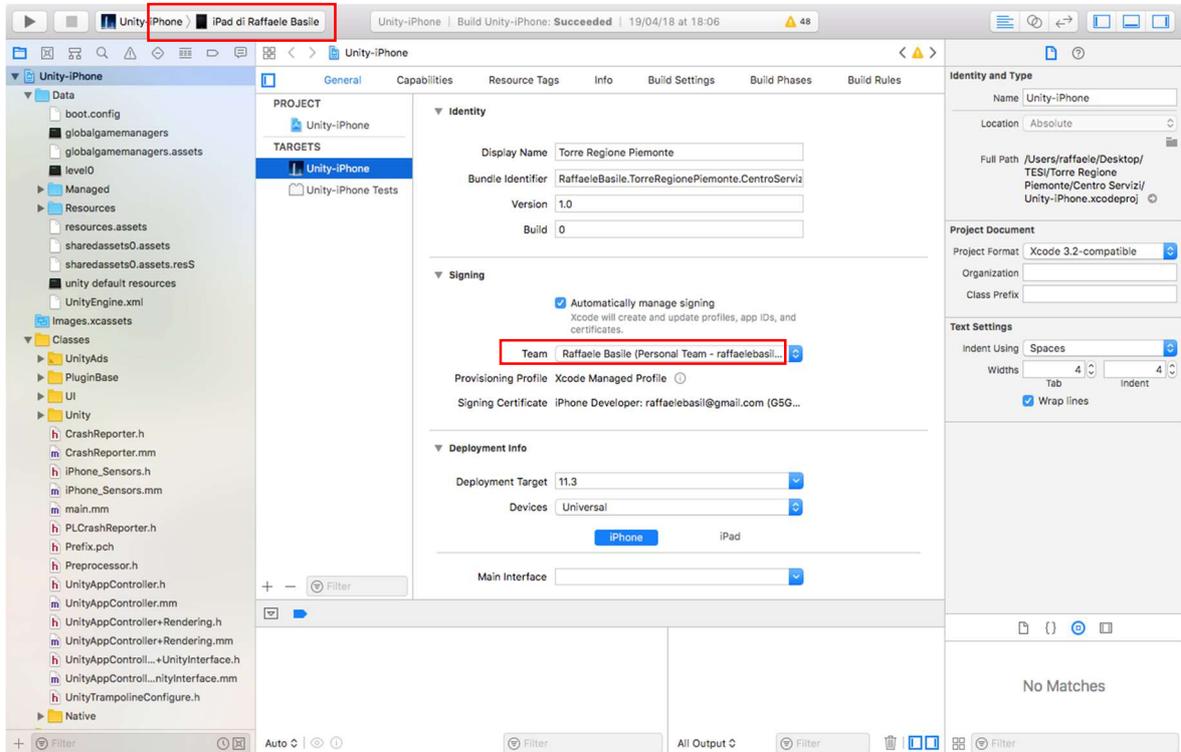


Figura 64: Schermata di Xcode

Fatto ciò, basta cliccare sul tasto “Play” e l’applicazione verrà installata.

Una volta installata l’applicazione verrà eseguita in automatico.

Tramite la scheda “Debug Navigator” è possibile andare a controllare le risorse utilizzate dall’applicazione in fase di esecuzione della stessa sul dispositivo.

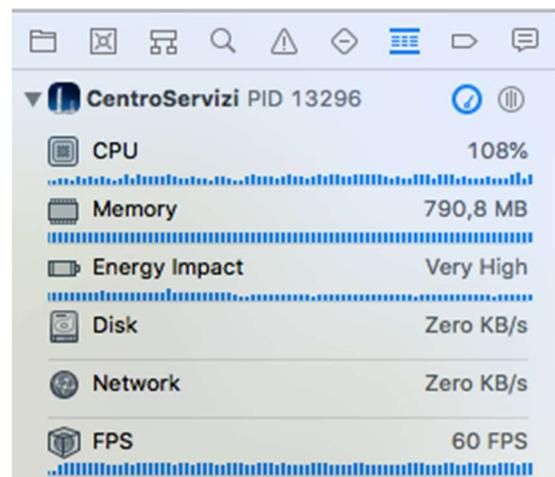


Figura 65: Statistiche di Debug

Questo pannello è stato molto utile per capire quanto comprimere le mesh all'inizio del lavoro in quanto si è monitorato costantemente la variazione di utilizzo di RAM a seconda dei vari rapporti di compressione delle mesh.

Il risultato migliore è il seguente come mostrato in figura. Come si può notare, l'applicazione mette a dura prova il dispositivo. Si è notato che dopo circa 10 minuti di utilizzo consecutivo il dispositivo si surriscalda molto. Ciò è dovuto al grande



Figura 67: QR Code del modello in scala 1:10

impatto che questa applicazione ha sulla batteria. Facendo altri test però si è arrivati alla conclusione che a mettere sotto sforzo la batteria non sia il modello di per sé, ma l'ARKit. Infatti

sono stati fatti altri test con un semplice cubo generico, e l'impatto sulla batteria non è cambiato. Di seguito è riportato il grafico dettagliato dell'utilizzo della CPU. Nell'allegato 12.5 è possibile visualizzare anche i grafici relativi alla RAM, batteria e telecamera (FPS).



Figura 66: Applicazione installata sull'iPhone7

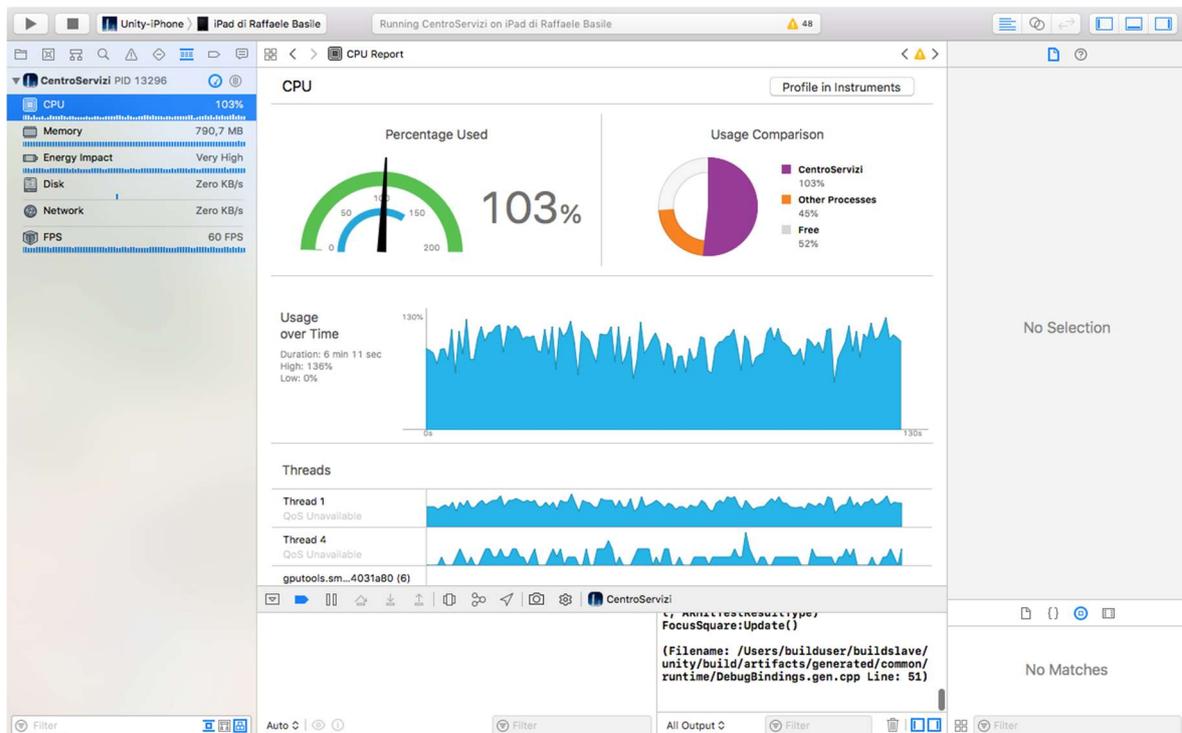


Figura 68: Utilizzo CPU

#### 6.2.3.4 Aggiunta di tasti per la visualizzazione

Come visto nell'ultimo video di prova, non c'era ancora la possibilità di scegliere cosa visualizzare del modello. Ciò risulta essere molto scomodo in quanto, pensando al risultato finale ovvero a quando si andrà nella struttura reale, si vorranno vedere solo gli impianti e non anche le strutture o l'architettonico. Per ovviare a ciò sono stati introdotti dei pulsanti figli nella gerarchia dell'elemento denominato "Canvas".

Per creare un pulsante basta cliccare con il tasto destro su uno spazio vuoto della gerarchia e selezionare UI > Botton. Nello specifico sono stati creati dei pulsanti che permettono di visualizzare e nascondere i singoli modelli importati. Per permettere di attivare o disattivare i modelli è stato creato un breve script in linguaggio C# denominato "Enable Object".

Creati i singoli pulsanti e lo script per la visualizzazione, si è proceduto al posizionamento di questi sullo schermo del dispositivo mobile.

Una volta creato il primo pulsante, Unity3D crea in automatico quello che sarà l'ingombro dello schermo del dispositivo, che, a seconda che sia un iPhone o un iPad, verrà automaticamente scalato. Ciò avviene anche in caso di rotazione del dispositivo.

Una volta creati i vari pulsanti, è stato modificato il testo contenuto andando ad inserire quello desiderato.

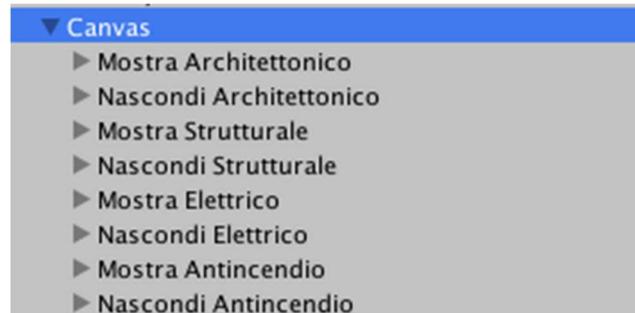


Figura 70: Pulsanti creati

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EnableObject : MonoBehaviour {
6     public void Enable_Object () {
7         gameObject.SetActive (true);
8     }
9 }
10 // Use this for initialization
11 void Start () {
12 }
13 }
14
15 // Update is called once per frame
16 void Update ()
17 {
18 }
19 }
20 }
21 }
```

Figura 69: Script per la visualizzazione dei modelli

Il testo di ogni singolo pulsante è figlio del pulsante stesso all'interno della gerarchia. Prima di collegare lo script ai singoli pulsanti bisogna selezionare il pulsante “+” nella sezione “OnClick” nell’Inspector.

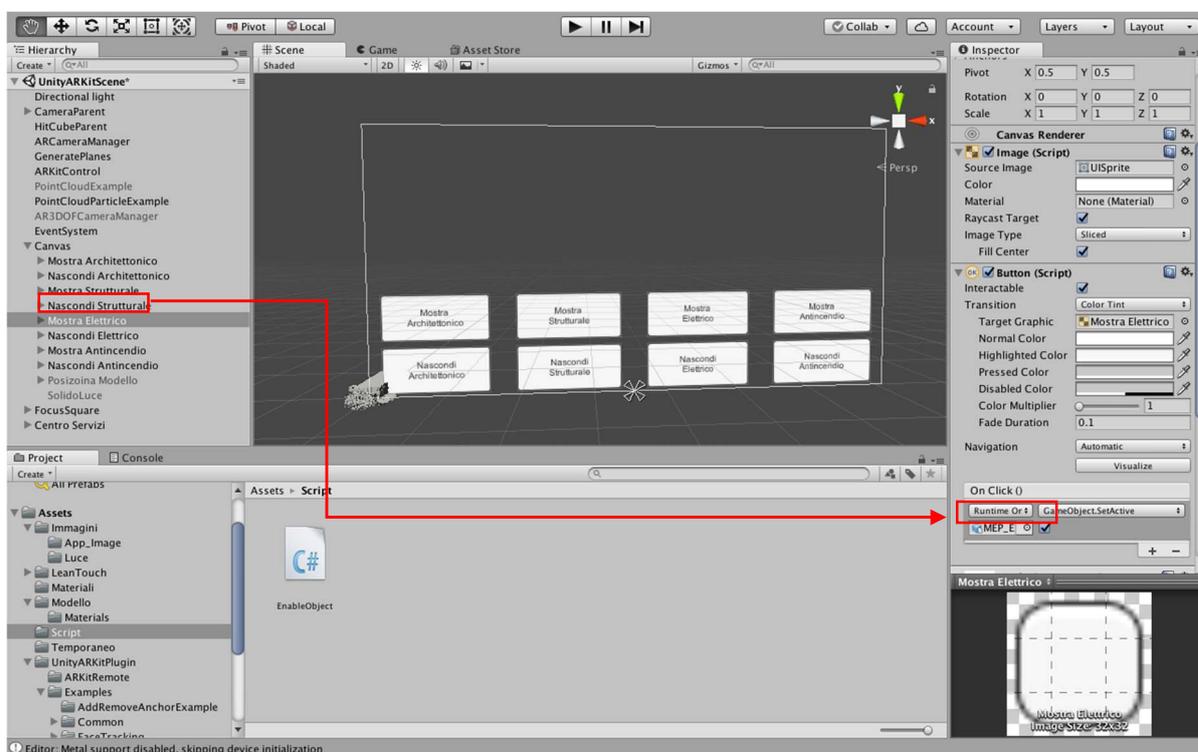


Figura 71: Collegamento dei pulsanti alle funzioni desiderate

Una volta aggiunta la funzione, è stato collegato il modello desiderato: la spunta è stata attivata nel caso si desiderasse far visualizzare il modello quando quel pulsante viene premuto; viceversa la spunta viene disattivata. Infine è stato scelto lo script da collegare a quel tasto cliccando su Function > GameObject > SetActive (bool). Questo procedimento è stato fatto per tutti e 8 i pulsanti creati.

Fatto ciò è sorto un problema: quando si tocca lo schermo del dispositivo, l’applicazione riaggiorna l’origine del modello. Ciò implica che ogni qualvolta si andava a disattivare uno dei 4 modelli presenti, l’intero modello si spostava. Ciò è stato ritenuto inammissibile, pensando allo scopo finale dell’applicazione, in quanto ciò avrebbe comportato un’alta probabilità di commettere errori. Per ovviare a questo problema si è andato a modificare lo script “UnityARHitTestExample” andando a implementare una parte di codice che sostanzialmente andasse a dire all’applicazione di non riconoscere come tocco quello dello schermo quando si va a toccare i pulsanti. Si riporta di seguito uno stralcio del codice. L’intero script è presente all’allegato 12.6.

```

7 public class UnityARHitTestExample : MonoBehaviour
8 {
9     public Transform m_HitTransform;
10    public float maxRayDistance = 30.0f;
11    public LayerMask collisionLayer = 1 << 10; //ARKitPlane Layer
12
13
14
15    bool HitTestWithResultType (ARPoint point, ARHitTestResultType resultTypes)
16    {
17        List<ARHitTestResult> hitResults = UnityARSessionNativeInterface.GetARSessionNativeInterface ().HitTest (point, resultTypes);
18        if (hitResults.Count > 0) {
19            foreach (var hitResult in hitResults) {
20                Debug.Log ("Got hit!");
21                m_HitTransform.position = UnityARMatrixOps.GetPosition (hitResult.worldTransform);
22                m_HitTransform.rotation = UnityARMatrixOps.GetRotation (hitResult.worldTransform);
23                Debug.Log (string.Format ("x:{0:0.#####} y:{1:0.#####} z:{2:0.#####}", m_HitTransform.position.x, m_HitTransform.position.y, m_HitTransform.position.z));
24                return true;
25            }
26        }
27        return false;
28    }
29
30
31
32    private bool IsPointerOverUIObject(){
33        PointerEventData eventDataCurrentPosition = new PointerEventData(EventSystem.current);
34        eventDataCurrentPosition.position = new Vector2(Input.mousePosition.x, Input.mousePosition.y);
35        List<RaycastResult> results = new List<RaycastResult>();
36        return results.Count > 0;
37    }
38
39    // Update is called once per frame
40    void Update () {
41
42        #if UNITY_EDITOR //we will only use this script on the editor side, though there is nothing that would prevent it from working on device
43        if (Input.GetMouseButtonDown (0)) {
44            Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);
45            RaycastHit hit;
46
47            //we'll try to hit one of the plane collider gameobjects that were generated by the plugin
48            //effectively similar to calling HitTest with ARHitTestResultType.ARHitTestResultTypeExistingPlaneUsingExtent
49            if (Physics.Raycast (ray, out hit, maxRayDistance, collisionLayer)) {
50                //we're going to get the position from the contact point
51                m_HitTransform.position = hit.point;
52                Debug.Log (string.Format ("x:{0:0.#####} y:{1:0.#####} z:{2:0.#####}", m_HitTransform.position.x, m_HitTransform.position.y, m_HitTransform.position.z));
53
54                //and the rotation from the transform of the plane collider
55                m_HitTransform.rotation = hit.transform.rotation;
56            }
57        }
58        #else
59        if (Input.touchCount > 0 && m_HitTransform != null)
60        {
61            var touch = Input.GetTouch(0);
62            if ((touch.phase == TouchPhase.Began || touch.phase == TouchPhase.Moved) && !IsPointerOverUIObject() && !EventSystem.current.IsPointerOverGameObject(0))
63

```

Test Results Errors Tasks

Figura 72:Unity AR Hit Test Example script modificato

### 6.2.3.5 Un modo migliore di posizionare il modello: FocusSquare

Arrivati a questo punto è sorto un ulteriore problema: Il posizionamento del modello risultava essere troppo approssimativo e non era possibile controllare la rotazione dello stesso attorno al proprio asse “Y”, inoltre, toccando i vari elementi sullo schermo per far apparire le informazioni collegate, molto spesso capitava che si toccasse invece lo schermo, con la conseguenza che il modello si riposizionava.

La visualizzazione delle informazioni dei singoli oggetti sarà argomento del prossimo paragrafo.

Ciò è stato ritenuto inammissibile perché c’era il rischio di incorrere troppo spesso in errore. Per ovviare a questo problema è stata utilizzata una funzionalità dell’ARKit chiamata FocusSquare.

Per aggiungere questa componente al progetto si è creato nel pannello delle gerarchie un “EmptyObject” rinominato FocusSquare per chiarezza.

All’interno di quest’ultimo con un drag and drop sono stati inseriti due componenti reperibili all’interno degli Assets chiamati “FocusSquareFinding” e “FocusSquareFound”. Successivamente è stato creato un altro Empty Object rinominato “Centro Servizi” all’interno del quale sono stati trascinati i singoli modelli importati in precedenza.

Una volta fatto ciò, si è andati ad aggiungere uno script nell’Inspector dell’ Empty Object creato in precedenza e rinominato “FocusSquare”.

Lo script aggiunto denominato “Focus Square (script)” è reperibile all’interno degli asset già importati con ARKit. Una volta importato lo script, sono stati fatti gli opportuni collegamenti come da immagine. Per quanto riguarda la voce “Button” è stata utilizzata per collegarci un pulsante presente sullo schermo che permette di dire all’applicazione quando l’utente vuole posizionare il modello.

Per la sua creazione è stato seguito un passaggio analogo a quanto visto prima.

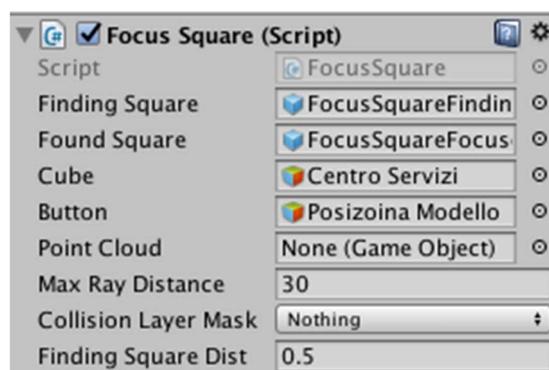


Figura 73: Focus Square Script

Una volta creato, sono state create due voci come da foto.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EnableObject : MonoBehaviour {
6     public void Enable_Object () {
7         gameObject.SetActive (true);
8     }
9 }
10 // Use this for initialization
11 void Start () {
12 }
13 }
14
15 // Update is called once per frame
16 void Update ()
17 {
18 }
19 }
20 }
21
```

Figura 75: Enable Object Script

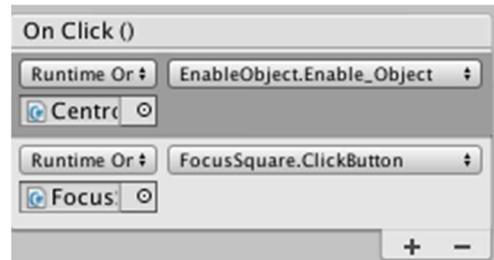


Figura 75: Script collegati al tasto "Posiziona Modello"

Nella seconda funzione è stato collegato lo script del Focus Square, mentre nel primo è stato collegato uno script che permette di visualizzare il modello solo dopo che questo è stato posizionato. Senza questo script comunque l'applicazione funzionerebbe, ma anche in fase di posizionamento del modello questo sarebbe visibile, disturbando l'utente nella fase di posizionamento. Il risultato ottenuto è riportato nel video collegato al codice QR presente in figura 76.



Figura 76: Risultati ottenuti

### 6.2.3.6 Collegamento delle informazioni agli elementi

Il passo successivo è stato quello di inserire delle informazioni collegate ai singoli elementi impiantistici. A titolo esemplificativo sono state inserite le informazioni circa la manutenzione delle sole luci e dei diffusori sonori.

In quanto non è stato fornito il piano di manutenzione, a titolo esemplificativo è stata utilizzata una documentazione trovata su internet.

In particolare si sono usati i dati relativi al piano di manutenzione programmata degli uffici della Direzione provinciale di Benevento.

Di seguito sono riportate le informazioni riportate nel suddetto documento e che sono state utilizzate nel presente progetto.

Corpi illuminanti luce normale per illuminazione interna ed esterna	Verifica e controllo funzionamento dei punti luce e apparecchiature varie dei circuiti di illuminazione (quadretti di illuminazione e comando, interruttori, interruttori differenziali, tubazioni e canaline protettive, fusibili, schermi protettivi, prese luce, collegamenti di terra, interruttori crepuscolari, ecc.)	1 mese
	Controllo ed eventuale sostituzione lampade non funzionanti	1 settimana
	Pulizia e controllo fissaggio	6 mesi
	Controllo funzionalità sistemi regolazione e controllo circuiti illuminazione	3 mesi
	Verifica corretta regolazione dispositivi di protezione	3 mesi
Diffusione sonora	Controllo: Controllo generale Ispezione a vista Controllare l'efficienza dei dispositivi di diffusione sonora contro l'apertura e l'asportazione. Verificare l'efficienza dello stato di carica della batteria di alimentazione.	3 mesi

Figura 77: Date della manutenzione programmata reperiti su internet

Una volta reperite le informazioni necessarie, si è proceduto alla creazione di un piccolo menu dedicato per ogni elemento. È stato creato un “canvas” settato con il render mode in “World Space” in modo tale che andasse ad occupare una dimensione ben precisa invece che tutto lo schermo (come avviene ad esempio per il canvas contenente il Main Menu o il Setting Menu)

Il menu si presenta come in figura 74.

Una volta creato il menu sono sorti 2 problemi:

- Rendere il menu “mobile” e non fisso in modo tale che esso risultasse sempre rivolto verso l’operatore. Ciò è stato fatto per pura comodità in quanto permette di poter visualizzare

agevolmente il menu da qualsiasi angolazione senza doversi collocare necessariamente in una posizione prestabilita.

- Registrare la data in cui si è fatta la manutenzione e in automatico restituire la data in cui bisognerà eseguire nuovamente la manutenzione, sempre sulla base dei dati precedentemente riportati.

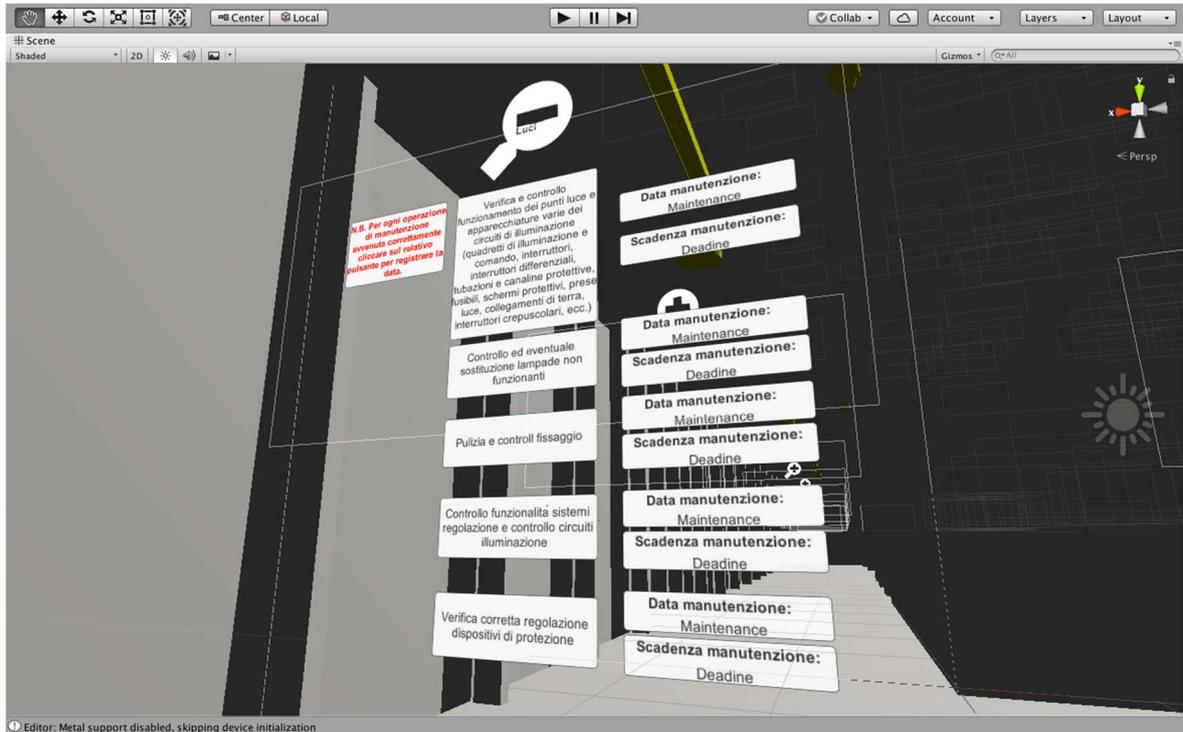


Figura 78: Menu dedicato per ogni elemento

Per ottemperare al primo punto è stato creato un semplice script il quale viene inserito all'interno di ogni Canvas e collegato alla Main Camera. All'interno dello script si dice al Canvas di ruotare la propria posizione in modo tale che il proprio asse X sia sempre ortogonale al piano della Main Camera.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class LookAtCamera : MonoBehaviour {
6
7     public Transform target;
8
9     void Update()
10    {
11        // Rotate the camera every frame so it keeps looking at the target
12        transform.LookAt(target);
13    }
14
15 }
16

```

Figura 79: LookAtCamera.cs

Il soddisfacimento del secondo punto invece è stato più complesso. È stato creato uno script denominato clock che permette di leggere la data e l'orario corrente direttamente dal dispositivo sul quale l'applicazione è in esecuzione e, in base a questa informazione, trascrive in corrispondenza dei pulsanti, la data attuale e quella di scadenza della manutenzione.

Di seguito è riportato uno stralcio dello script. Il codice intero è riportato all'allegato 12.7.

```

38 void Start() {
39
40     currentTime = System.DateTime.Now;
41
42     #region date converter
43     currentdateDAY= int.Parse(currentTime.ToString("dd"));
44     currentdateMONTH = int.Parse(currentTime.ToString("MM"));
45     currentdateYEAR = int.Parse(currentTime.ToString("yyyy"));
46     currentdate = currentdateDAY + (currentdateMONTH * 30) + (currentdateYEAR * 365);
47     Debug.Log(currentdate);
48     #endregion
49
50
51     dateMaintenance.text = PlayerPrefs.GetString(GetInstanceID() + ". " + "dateMaintenance" , currentTime.ToString("dd/MM/yyyy").ToString());
52     maintenanceDeadline.text = PlayerPrefs.GetString(GetInstanceID()+". " + "maintenanceDeadline" , deadlineTime.ToString("dd/MM/yyyy").ToString());

```

Figura 80:Stralcio dello script Clock.cs

Grazie a questo script è quindi possibile ottenere in automatico la data della prossima manutenzione per ogni elemento. Per far funzionare correttamente lo script, come già spiegato prima, sono stati collegati i button allo script e ad ogni button nella sezione Onclick() è stata selezionata la funzione denominata “Clock.maintenanceCommand()”.



Figura 81: Screenshot di un menu riguardante le luci

### 6.2.3.7 Diversi tipi di salvataggio

Una volta ottenuti i dati relativi alle date in cui avviene la manutenzione e le rispettive scadenze è sorta in automatico la necessità di archiviare tali informazioni.

L'archiviazione dei dati è risultata essere una delle fasi più difficili e complesse nella realizzazione dell'applicazione in quanto, a causa dell'elevato numero di elementi presenti nel progetto, bisognava cercare di dare un'adeguata flessibilità allo script affinché fosse il più versatile possibile.

Sono stati sperimentati 3 metodologie di salvataggio:

- PlayerPrefs
- SQLite
- MySQL

#### 6.2.3.7.a PlayerPrefs

Il primo metodo di salvataggio sperimentato è chiamato "PlayerPrefs".

Esso permette il salvataggio automatico delle date in cui avviene la manutenzione e della loro relativa scadenza e il caricamento automatico dei medesimi dati quando la scena viene aperta.

Questa funzione è strutturata in modo tale da poter salvare una stringa, un numero intero o un numero con la virgola (float). All'interno della parentesi che la seguono si trovano due valori separati da una virgola. Il primo è il nome del file che si andrà a creare quando il dato verrà salvato, il secondo è il dato che si vuole salvare. La limitazione di questa funzione è che può salvare solo un dato per file. Per ovviare a questo problema si è lavorato sul nome del file di salvataggio. Infatti si è deciso di denominare il file "GetInstanceID() + "\_" + "maintenance"" per quanto riguarda la data in cui avviene la manutenzione e "GetInstanceID() + "\_" + "maintenanceDeadline"" la relativa scadenza.

Si è deciso di anteporre l'ID dell'elemento in modo tale da andare a creare un nuovo file per ogni elemento. Se non si inserisse l'ID all'interno del nome del file di salvataggio ogni volta che si andrebbe a salvare un file, questo sovrascriverebbe il dato precedente in quanto aventi lo stesso nome. Per verificare quanto detto, si è scaricata un'applicazione che permettesse di esplorare i file di sistema dell'iPhone e si è potuto verificare la veridicità di quanto detto prima.

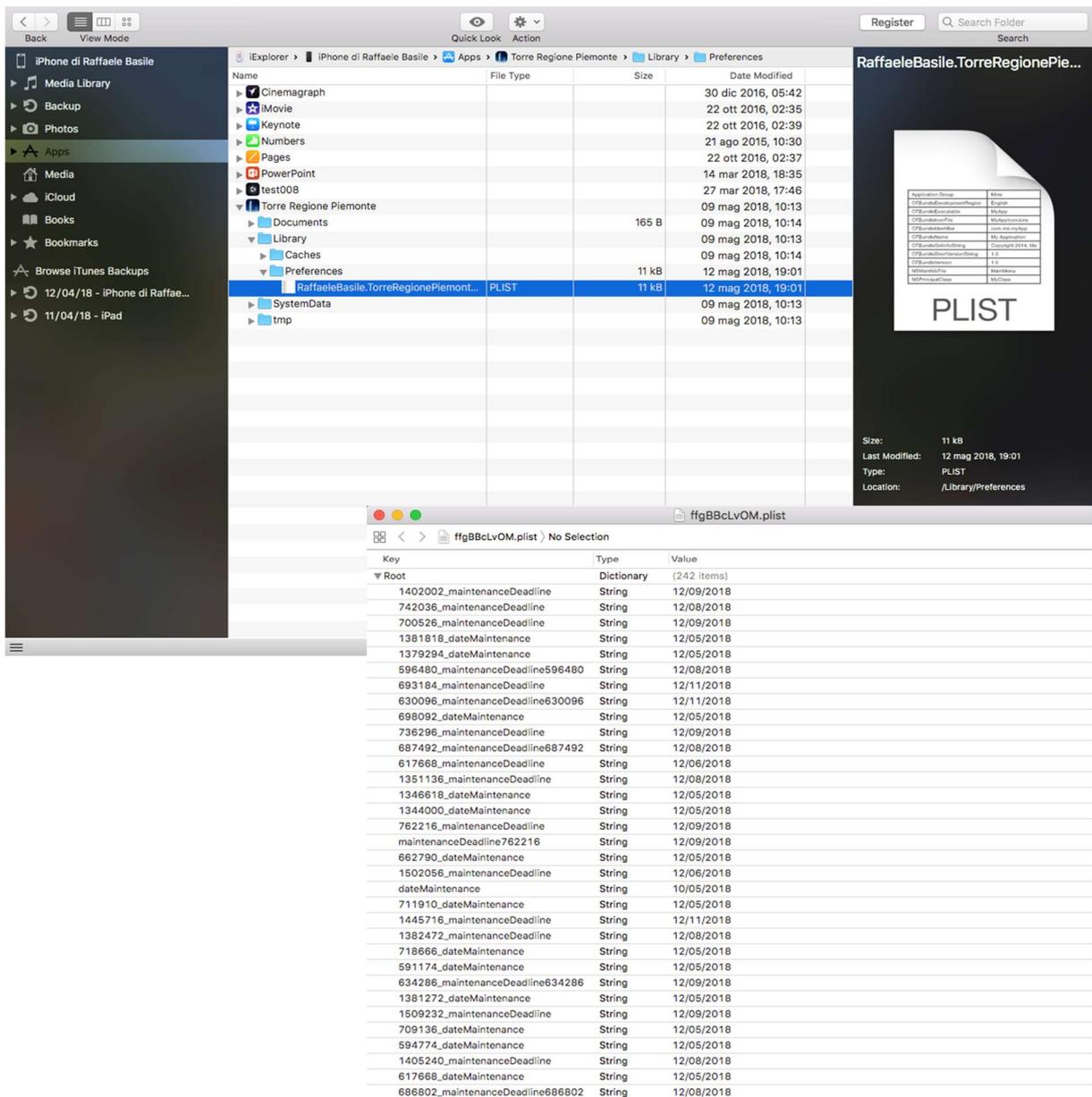


Figura 82: File salvati all'interno dell'iPhone utilizzato come test dell'applicazione

Questo metodo di salvataggio non è risultato essere soddisfacente in quanto la procedura per poter accedere ai dati salvati è troppo macchinosa e non ne permette il loro riutilizzo.

Si è provato infatti ad aprire questo file di salvataggio con programmi in grado di gestire dei database, ma né Access né Excel, importando il file in formato CSV, erano in grado di leggere correttamente il documento.

Arrivati a questo punto si è deciso di adottare un'altra soluzione: SQLite.

#### 6.2.3.7.b SQLite

SQLite è una libreria software scritta in linguaggio C che implementa un DBMS SQL di tipo ACID incorporabile all'interno di applicazioni. Il suo creatore, D. Richard Hipp, lo ha rilasciato nel pubblico dominio, rendendolo utilizzabile quindi, senza alcuna restrizione.



*Figura 83: SQLite Logo*

Permette di creare una base di dati (comprese tabelle, query, form, report) incorporata in un unico file, come nel caso dei moduli Access di Microsoft Office e Base di OpenOffice.org e Libre Office; analogamente a prodotti specifici come Paradox o Filemaker.

SQLite non è un processo standalone utilizzabile di per sé, ma può essere incorporato all'interno di un altro programma. È utilizzabile con il linguaggio C/C++, ed esistono binding anche per altri linguaggi, in particolare Tcl. È inoltre stato integrato nella versione 5 di PHP, consentendo a tale popolare linguaggio di disporre di un altro RDBMS indipendentemente dalla presenza di MySQL. Viene utilizzato in Mozilla Firefox e Seamonkey per memorizzare i segnalibri, la cronologia di navigazione ed altre informazioni.

Per l'utilizzo di SQLite si è innanzitutto scaricato SQLite Manager come estensione di Firefox.

Poiché la versione utilizzata è risultata essere obsoleta rispetto all'attuale versione del browser, si è installata la versione 27.0 di quest'ultimo.

Una volta fatto ciò, cliccando su strumenti compare SQLite Manager. Si è scelto di salvare il DB formato ":sqlite" nella cartella denominata "StreamingAssets" presente nella cartella "Assets" dell'applicazione. Si vedrà in seguito il motivo di questa scelta.

Aprendo il DB dall'SQLite Manager, cliccando col tasto destro del mouse su "Tables" si è creata una nuova tabella.

All'interno della schermata che compare si sono andati ad inserire i cambi da cui sarà composto il DataBase.

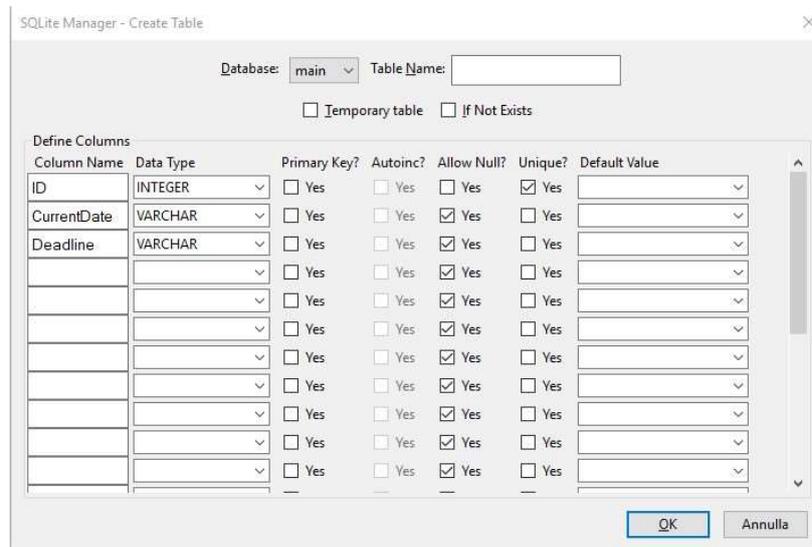


Figura 84: Maintenance Table

Sono stati inseriti i seguenti campi:

- ID: di tipo INTEGER. Andrà a contenere l'id del pulsante della manutenzione. È stato impostato di tipo unico in quanto l'id degli elementi è univoco.
- CurrentDate: di tipo VARCHAR. Andrà a contenere la data in cui viene eseguita la manutenzione.
- Deadline: di tipo VARCHAR. Andrà a contenere la data in cui scade la manutenzione.

Una volta creata la tabella sono stati fatti dei test all'interno di SQLite per verificare che il database funzionasse correttamente e per acquisire i comandi di input da dare al database stesso per ottemperare a diversi comandi che verranno ripresi dallo script successivamente.

Il primo comando utilizzato è stato il seguente: `SELECT * FROM Maintenance`

Questo comando dice al database di mostrare a video tutto il contenuto della tabella denominata “Maintenance”

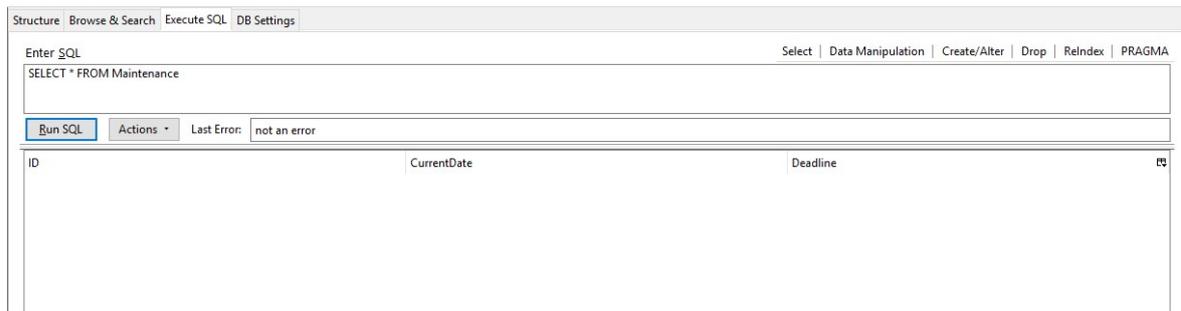


Figura 85: *SELECT \* FROM Maintenance*

Ovviamente al momento la tabella è vuota in quanto è stata appena creata. Ma ciò è servito da verifica per constatare che i campi fossero stati creati.

Successivamente si è provato ad inserire dei valori casuali, sempre come test.

Il comando utilizzato è stato: `INSERT INTO Maintenance(ID,CurrentDate,Deadline) VALUES (1002,"31/05/2018","31/06/2018");`

Questo comando dice al database di inserire nella tabella Maintenance dei valori nelle colonne ID, CurrentDate e Deadline. I valori sono specificati all’interno di VALUES.

SQLite restituisce come output “not an error” che sta ad indicare che l’aggiunta dei valori è andata a buon fine. A questo punto per visualizzare il contenuto del database basta ripetere il primo comando visto in precedenza. Il risultato è il seguente.

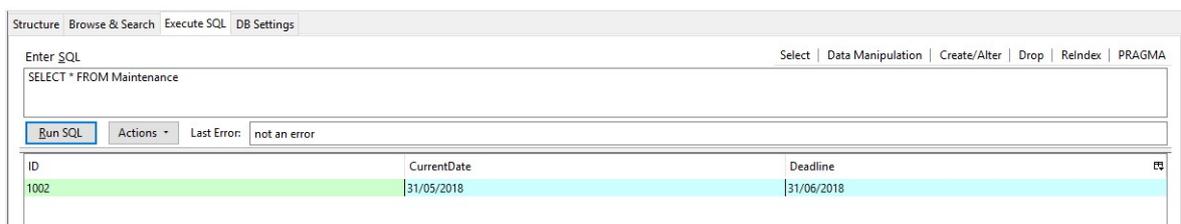


Figura 86: *Contenuto del database*

L’ultimo comando utilizzato è stato quello per cancellare gli elementi dal database: `DELETE FROM Maintenance WHERE ID = “1002”;`

Questo comando permette di cancellare tutta la riga dove nella colonna ID è presente un valore pari a 1002. Volendo si potrebbe effettuare la ricerca per data in modo tale da andare ad eliminare tutti gli ID i quali presentano una data di manutenzione pari ad un determinato giorno. In quel caso si otterrebbe un’eliminazione di più righe contemporaneamente.

Nel presente caso è stata utilizzata l'eliminazione per ID.

Una volta impostato il database, ed eliminate le righe di prova, si è passato alla creazione dello script. All'interno dello script Clock.cs visto in precedenza, sono state aggiunte 2 funzioni, una per la scrittura del database e una per la cancellazione delle righe da aggiornare.

```
144 #region SQLite Manager
145 private string connectionString;
146
147 public void Delete(){
148     DeleteData (GetInstanceID());
149 }
150
151 }
152
153 private void InsertData(int ID, string CurrentDate, string Deadline, int Counter){
154     using (IDbConnection dbConnection = new SQLiteConnection (connectionString)) {
155         dbConnection.Open ();
156         using (IDbCommand dbCmd = dbConnection.CreateCommand ()) {
157             string sqlQuery = String.Format ("INSERT INTO Maintenance (ID,CurrentDate,Deadline,Counter) " +
158                 "VALUES(\{0}\",\{1}\",\{2}\",\{3}\")",
159                 ID, CurrentDate, Deadline, Counter);
160
161             dbCmd.CommandText = sqlQuery;
162             dbCmd.ExecuteNonQuery ();
163             dbConnection.Close ();
164         }
165     }
166 }
167
168 }
169
170 }
171
172 }
173
174 private void DeleteData(int ID){
175     using (IDbConnection dbConnection = new SQLiteConnection (connectionString)) {
176         dbConnection.Open ();
177         using (IDbCommand dbCmd = dbConnection.CreateCommand ()) {
178             string sqlQuery = String.Format ("DELETE FROM Maintenance WHERE ID =\{0}\\"",ID);
179             dbCmd.CommandText = sqlQuery;
180             dbCmd.ExecuteNonQuery ();
181             dbConnection.Close ();
182         }
183     }
184 }
185
186 }
187
188 }
189
190 }
191 #endregion
192
```

Figura 87: Script per la gestione del database con SQLite

Sono state create 3 funzioni:

- Delete()
- InsertData()
- DeleteData()

Nella fattispecie InsertData si occupa dell'inserimento dei dati all'interno del database.

La funzione è stata impostata in modo tale che inserisse l'ID del menu di manutenzione selezionato, la data di manutenzione e la scadenza. Quest'operazione avviene tramite una stringa chiamata `sqlQuery` che assume il contenuto del comando visto in precedenza per l'inserimento di dati direttamente da SQLite. La funzione `DeleteData` funziona in maniera analoga ma l'unico valore di input di cui necessita è l'ID in quanto una volta individuato l'ID, elimina l'intera riga corrispondente ad esso.

La funzione `DeleteData` è stata inclusa nella funzione `Delete` in modo tale da acquisire tramite il comando "`GetInstanceID()`", l'id del pulsante premuto.

L'eliminazione della riga è fondamentale in quanto essendo l'id una chiave primaria, non ammette un'altra istanza con il medesimo id; ciò significa che quando si andrà ad aggiornare una data di manutenzione per lo stesso elemento, prima bisognerà eliminare la riga contenente la medesima manutenzione avvenuta in precedenza.

Questa soluzione presenta alcune limitazioni, motivo per le quali si è scelto di adottare un'altra soluzione per l'archiviazione dei dati.

Innanzitutto i dati sono salvati localmente, dunque, nel caso in cui in loco andassero più persone contemporaneamente ed utilizzassero l'applicazione in diverse zone dell'edificio, questi otterrebbero due database con informazioni salvate differenti e senza la possibilità di "unire" i due database. Inoltre potrebbe esserci il problema che gli ID del medesimo elemento siano diversi su dispositivi diversi. Dato l'elevato numero di incertezze su questa soluzione, si è optato per un'archiviazione in cloud sfruttando MySQL e Altrivista.

#### 6.2.3.7.c MySQL

Per risolvere il problema dell'utilizzo contemporaneo dell'applicazione da diversi dispositivi e per unificare i dati di salvataggio in un unico database si è scelta una soluzione in cloud. Si è scelto di utilizzare il sito Altrivista il quale permette di creare un sito online con la possibilità di creare un database e di collegarsi, tramite una connessione dati da dispositivo mobile, ad esso.

Per prima cosa è stato creato il sito web e denominato <http://tesingegneriaedilepolito.altervista.org>. Se si cerca questo sito tramite un browser non si riesce a trovare una vera e propria pagina internet in quanto non è stata creata alcuna pagina html ma soltanto il database. Dalla pagina di gestione del sito è stato creato innanzitutto il database.

Per riuscire a ottenere questo risultato, dato che le competenze informatiche richieste erano molto elevate, si è lavorato in collaborazione con uno studente di ingegneria informatica che ha aiutato il sottoscritto alla realizzazione degli script in php.

The screenshot shows a database management interface for a table named 'data'. The table has four columns: 'objID', 'menuID', 'dateA', and 'dateB'. The 'objID' column is the primary key. The interface includes a toolbar with options like 'Mostra', 'Struttura', 'SQL', 'Cerca', 'Inserisci', 'Esporta', 'Importa', and 'Operazioni'. Below the table structure, there are options to 'Visualizza per stampa', 'Proponi la struttura della tabella', and 'Move columns'. There is also a section for 'Indici' (Indexes) and a table of 'Informazioni' (Information) about the table's statistics.

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito	Extra	Azione
1	objID	varchar(160)	utf8mb4_general_ci		No	Nessuno		Modifica Elimina Primaria Unica Indice Spacial Testo completo Valori distinti
2	menuID	varchar(64)	utf8mb4_general_ci		No	Nessuno		Modifica Elimina Primaria Unica Indice Spacial Testo completo Valori distinti
3	dateA	date			No	Nessuno		Modifica Elimina Primaria Unica Indice Spacial Testo completo Valori distinti
4	dateB	date			No	Nessuno		Modifica Elimina Primaria Unica Indice Spacial Testo completo Valori distinti

Spazio utilizzato		Row statistics	
Dati	3.2 KIB	Formato	dinamico
Indice	8 KIB	Codifica caratteri	utf8mb4_general_ci
Totale	11.2 KIB	Righe	31
		Lunghezza riga	106 B
		Dimensione riga	371 B
		Creazione	Mag 31, 2018 alle 16:00
		Ultimo cambiamento	Giu 02, 2018 alle 12:55

Figura 88: Struttura del database

Nel database, denominato di default `my_tesingegneriaedilepolito`, è stata creata una tabella denominata `data`. All'interno di essa sono stati creati i campi che costituiscono il database:

- `objID`
- `menuID`
- `dateA`
- `dateB`

I primi due elementi sono chiavi primarie, ciò significa che il database non ammette due righe i cui dati siano identici. Quando ciò avviene il dato viene sovrascritto: è il caso in cui viene effettuata una nuova manutenzione sul medesimo oggetto.

`dateA` è la data attuale in cui avviene la manutenzione mentre `dateB` la data di scadenza.

Una volta creato il database sono stati creati 3 file php che permettessero di far comunicare il database con l'applicazione. I file sono presenti all'allegato 12.9.

Una volta creati i file di collegamento si è modificato il file presente nell'applicazione affinché potesse connettersi e comunicare con il database. Gli script sono stati totalmente riscritti in quanto se prima tutti i menu erano presenti all'interno dell'applicazione, con la nuova organizzazione degli script vengono generati e distrutti ogni qualvolta il menu di un determinato elemento viene visualizzato o nascosto. Questo riadeguamento dei menu è stato necessario poiché la presenza di troppi menu caricati contemporaneamente iniziava ad essere gravoso per la RAM del dispositivo.

Gli script creati sono stati denominati objManager, responsabile della creazione e distruzione Runtime dei menu, e MenuManager, il quale si occupa della comunicazione tra applicazione e database in Cloud.

```
--
13 public class ObjManager : MonoBehaviour {
14
15     public string nameObject;
16     public GameObject Menu;
17     private GameObject menuInstance;
18     public bool ExistingYet;
19
20     private void Start(){
21         ExistingYet = false;
22     }
23
24     private void Update()
25     {
26
27     }
28
29     void OnMouseDown() {
30         nameObject = gameObject.name;
31
32         if (ExistingYet == false) {
33             ExistingYet = true;
34
35             Vector3 localpos = transform.position;
36             localpos.y -= 1.50f;
37             Debug.Log("NEW POS: "+ localpos);
38
39
40
41             menuInstance = Instantiate(Menu, localpos,transform.rotation);
42             menuInstance.transform.parent = gameObject.transform;
43
44         } else {
45             ExistingYet = false;
46             Destroy(menuInstance);
47         }
48     }
49 }
50
51 }
52 }
53 }
```

Figura 89: objManager.cs

```

38 Load(ObjName, MenuName, (dateA, dateB) => {
39     Debug.Log("loading " + ObjName + " " + MenuName);
40     // Clock.maintenanceDeadline.text = dateA.ToString("dd/MM/yyyy");
41     // dateA e dateB contengono le due date per oggetto menu
42     // questa porzione di codice viene eseguita se è tutto ok
43     dateMaintenance.text = dateA.ToString("dd/MM/yyyy");
44     maintenanceDeadline.text = dateB.ToString("dd/MM/yyyy");
45 }, () => {
46     //questa porzione di codice invece viene eseguita al posto di quella sopra
47     //nel caso in cui il recupero non abbia avuto successo
48     Debug.Log("error-not found");
49     dateMaintenance.text = maintenanceDeadline.text = "Manutenzione mai avvenuta";
50
51
52 });
53
54 }
55
56 // Update is called once per frame
57 void Update () {
58
59 }
60
61 public void maintenanceCommand()
62 {
63     currentTime = DateTime.Now;
64     deadlineTime = currentTime.AddMonths(howMuchTime);
65     dateMaintenance.text = maintenanceDeadline.text = "Impostazione..";
66     Save(ObjName, MenuName, currentTime, deadlineTime, (success) =>
67     {
68         if (success)
69         {
70             dateMaintenance.text = currentTime.ToString("dd/MM/yyyy");
71             maintenanceDeadline.text = deadlineTime.ToString("dd/MM/yyyy");
72         }
73         else
74         {
75             dateMaintenance.text = maintenanceDeadline.text = "Errore!";
76         }
77     });
78
79 }
80

```

*Figura 90:Stralcio dello script MenuManager.cs*

### 6.2.3.8 Impostazione del Main Menu

Una volta create tutte le funzionalità necessarie al corretto funzionamento dell'applicazione, sono state create altre quattro scene in modo tale da predisporre l'applicazione per quando saranno disponibili i modelli della torre, della corte interrata e del parcheggio.

È stata creata una scena contenente quattro pulsanti che, cliccandoci sopra, aprono la scena relativa al modello desiderato.

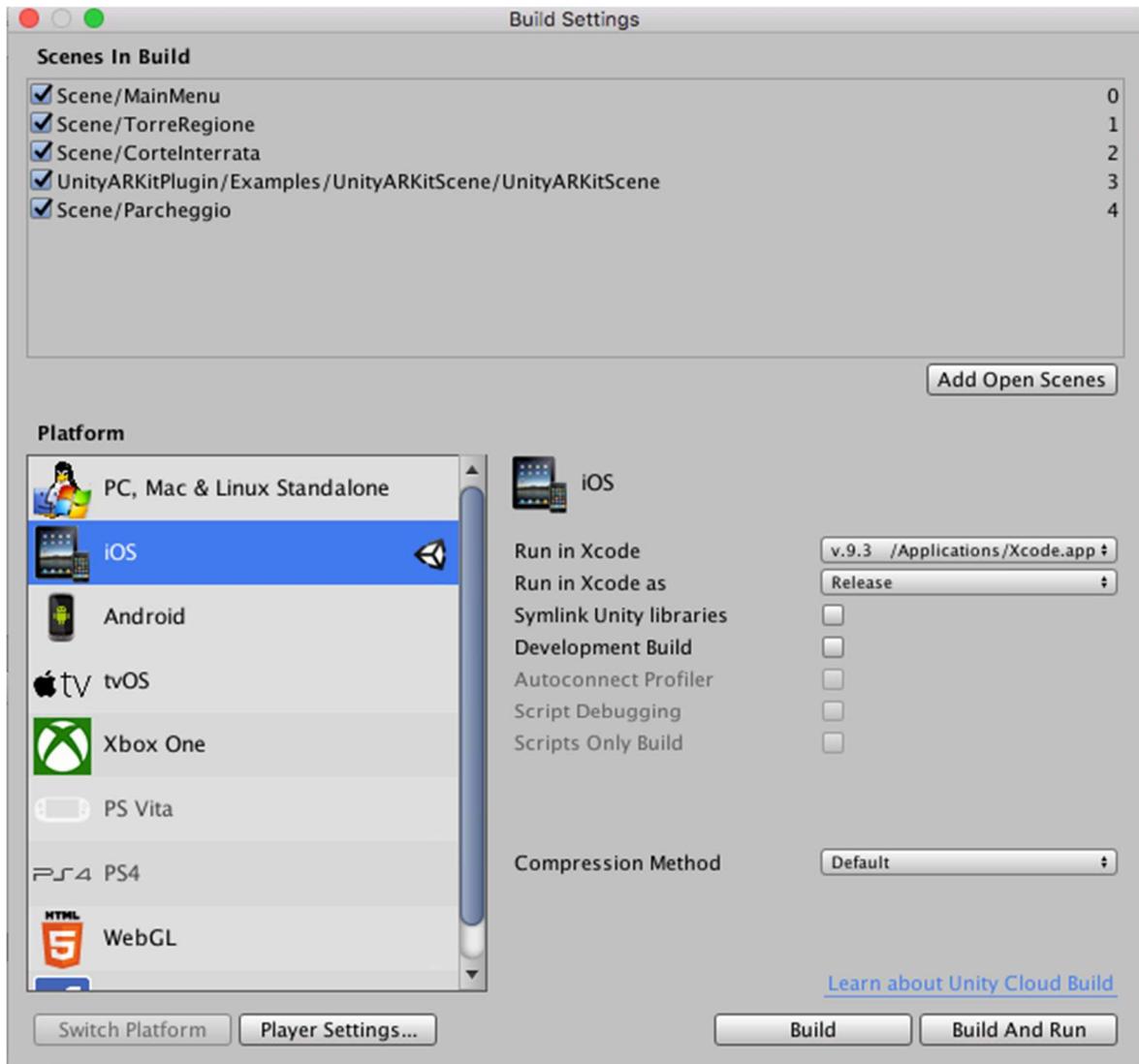


Figura 91: Gerarchia delle scene

Per fare ciò, dopo aver creato le scene, è stata impostata innanzi tutto la gerarchia delle scene. Per impostare questo ordine basta aprire il Build Settings e trascinare all'interno tutte le scene che si desidera inserire nell'applicazione. Il numero posto sulla destra delle scene è il loro numero nella gerarchia.

Questo numero sarà utile allo script utilizzato di seguito per permettere di switchare tra una scena e l'altra.

Questo script utilizza lo "SceneManagement" che permette di cambiare il livello della scena da caricare rispetto a quello di partenza. Si è utilizzato un unico script per caricare tutte le varie scene. Per fare ciò sono state create 4 funzioni denominate PlayTRP, PlayCI, PlayCS, PlayPRK. Ad ognuna di queste funzioni è stato dato un progressivo della scena da caricare in base alla loro posizione nell'elenco delle scene.

Uno script analogo è stato creato invece per tornare al Main Menu. Fondamentalmente invece di aggiungere un numero al livello, è stato sottratto.

L'utilizzo di più scene per visualizzare le singole parti del modello consente di non affaticare troppo la RAM del dispositivo in quanto, così facendo, viene caricata solo la zona del progetto dove ci si trova.

Una volta creato lo script è stata collegata la funzione ai singoli pulsanti.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class MainMenu : MonoBehaviour {
7
8     public void PlayTRP(){
9
10         SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex + 1);
11     }
12 }
13
14 public void PlayCI(){
15
16     SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex + 2);
17 }
18
19 public void PlayCS(){
20
21     SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex + 3);
22 }
23
24 }
25
26 public void PlayPRK(){
27
28     SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex + 4);
29 }
30 }
31 }
32 }
33 }
```

Figura 92: Script per cambiare scena

Analogamente, per tornare indietro dalle singole scene al Main Menu è stata utilizzata la funzione, ad esempio nel centro servizi, BackFromCS collegata in maniera analoga ad un tasto presente nella scena. Nelle scene dove non è ancora presente il modello compare semplicemente una scritta : “Not available yet” e cliccando su di essa si viene ricondotti al Main Menu.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class BackToMain : MonoBehaviour {
7
8     public void BackFromTRP(){
9
10         SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex - 1);
11     }
12
13     public void BackFromCI(){
14
15         SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex - 2);
16     }
17
18     public void BackFromCS(){
19
20         SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex - 3);
21     }
22
23     public void BackFromPRK(){
24
25         SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex - 4);
26     }
27
28 }
29
30
31
32
33

```

Figura 93: Script per tornare al Main Menu

Quando in futuro saranno presenti i modelli delle altre parti del progetto, questi verranno integrati nell’applicazione e sarà possibile accedere alle medesime funzionalità mostrate per il centro servizi anche per la torre, la corte interrata e i parcheggi. A questo punto è sorta la necessità di snellire le singole scene in quanto iniziavano ad essere presenti troppi pulsanti sulla schermata. Si è pensato di creare, all’interno della scena del centro servizi altri due menu chiamati rispettivamente “Settings” e “View Model”.

Accedendo al primo, cliccandoci sopra, compaiono i tasti per ruotare il modello, spostarlo sul piano Z-X e per tornare al Main Menu.

Accedendo al secondo, invece, compaiono i tasti per accendere o spegnere i modelli dello strutturale, architettonico, antincendio ed elettrico.

Per fare ciò basta andare nell’Inspector dei singoli tasti e nella sezione denominata “On click()”, basta trascinare dentro i tasti desiderati e tramite la funzione “SetActive()” decidere se far visualizzare o meno quel determinato tasto.

Infine è stato scaricato un Asset dall’Asset Store contenente icone in stile minimal le quali sono state assegnate ai singoli pulsanti per pura estetica.

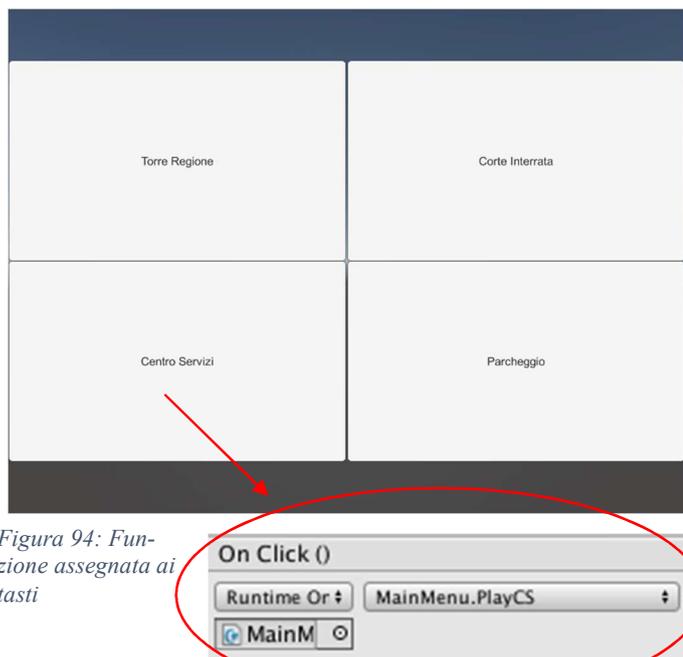


Figura 94: Funzione assegnata ai tasti



## 7 IMPORTAZIONE DELLE DATE AGGIORNATE IN REVIT

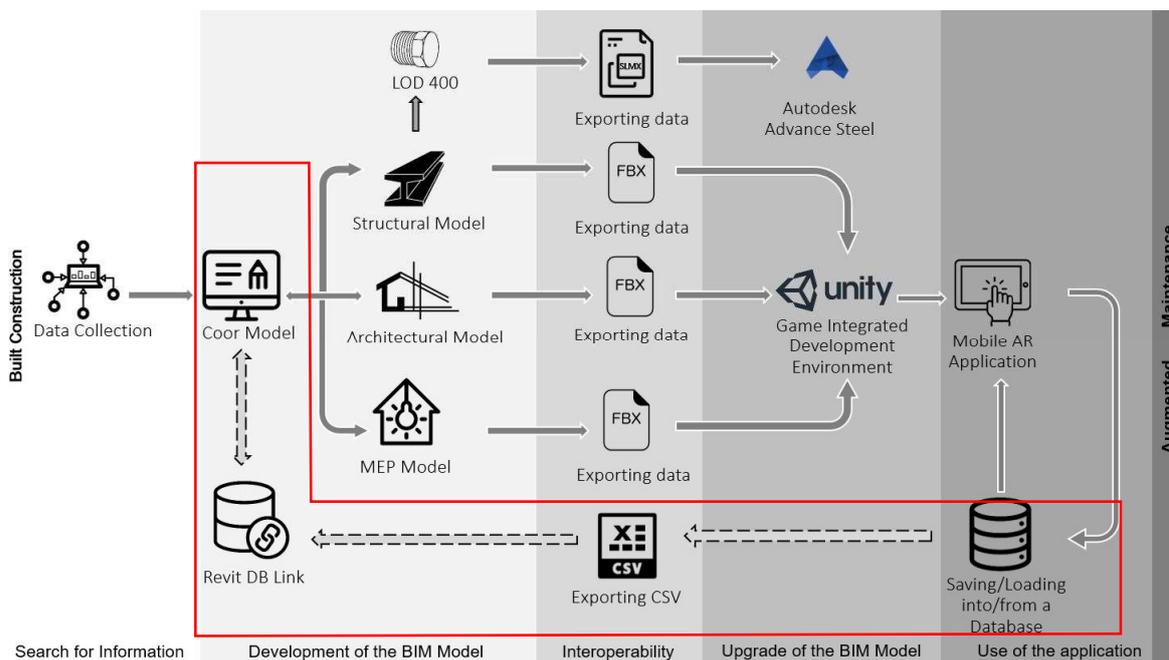


Figura 95: Posizione all'interno del workflow - Importazione dei dati all'interno del modello BIM

Una volta ottenute e salvate all'interno del database le informazioni riguardanti la manutenzione dei singoli elementi, si è passato alla loro importazione all'interno di Revit, il quale è stato il punto di partenza da cui è stato possibile sviluppare l'applicazione.

I passaggi sono stati:

- La creazione di parametri condivisi e di progetto all'interno del modello BIM riguardanti l'ID degli elementi e le relative manutenzioni;
- L'esportazione del database da Altervista, sito all'interno del quale è collocato il database;
- L'esportazione del database di Revit tramite DB Link;
- L'unione delle informazioni presenti nel database di Altervista con quelle del database di Revit e la loro successiva reimportazione all'interno di Revit e la creazione di un abaco delle manutenzioni.

### 7.1 La creazione dei parametri condivisi

Per prima cosa sono stati creati dei parametri condivisi all'interno dei quali verranno inseriti i dati relativi alla manutenzione importati dal database di Altervista.

Per la loro creazione basta andare nella scheda “Gestisci” e selezionare il comando “Parametri Condivisi”. Cliccando su “Crea” e scegliendo nome e destinazione del file, verrà creato un file di testo che andrà a racchiudere i parametri appena creati. Successivamente è stato creato il gruppo di parametri relativi all’illuminazione e al suo interno i parametri che effettivamente verranno visualizzati nelle proprietà degli elementi. Inoltre è stato creato un altro parametro condiviso chiamato “ID\_Elemento” la cui unica funzione è quella di fare da appoggio all’interno del database per poter visualizzare l’ID dell’elemento anche all’interno dell’abaco.

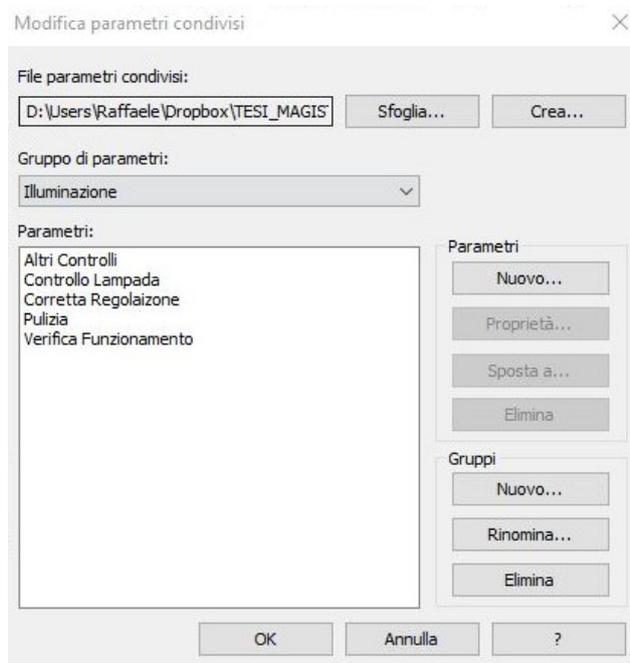


Figura 96: Elenco dei parametri condivisi relativi all’illuminazione

Una volta creati i parametri condivisi, sono stati creati quelli di progetto.

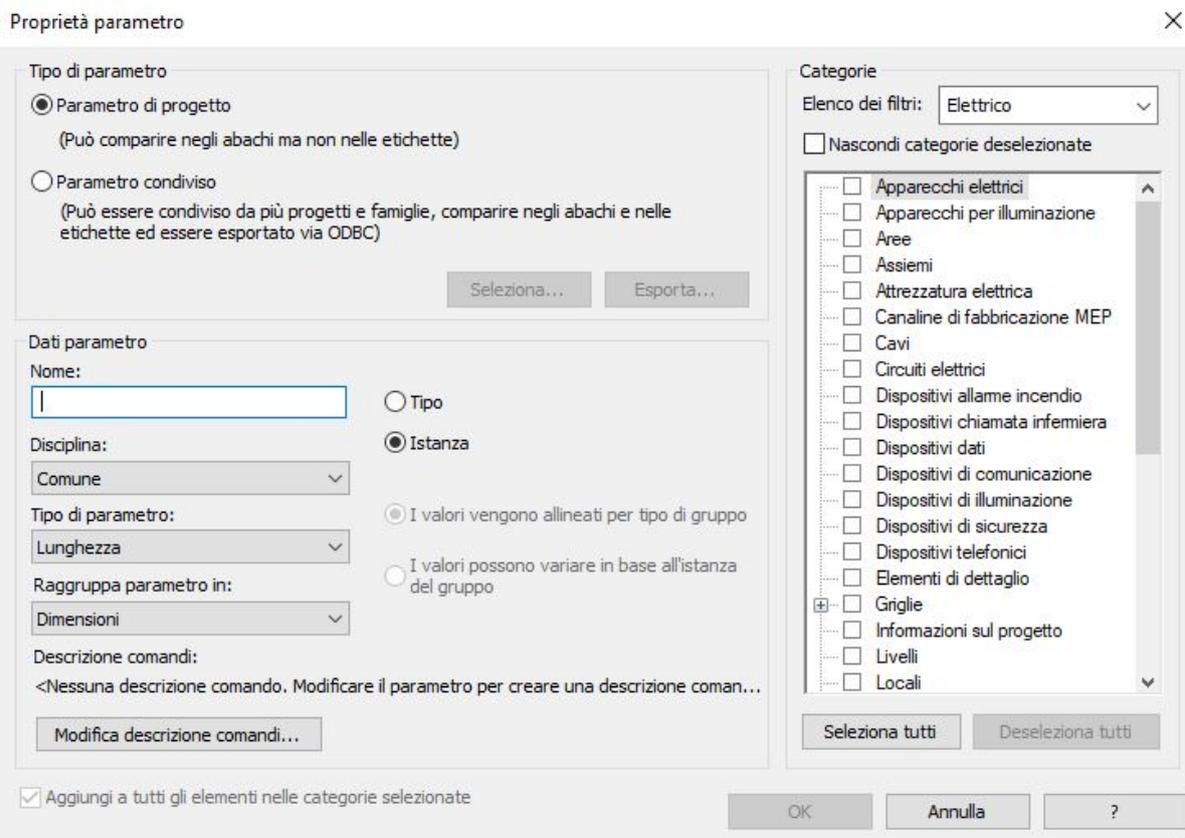


Figura 97: Tabella per la creazione dei parametri di progetto

Per quanto riguarda i parametri relativi all'illuminazione è stato selezionato solamente la voce "Apparecchi per l'illuminazione" mentre per quanto riguarda il parametro "ID\_Elemento" sono stati selezionate tutte le voci.

Una volta fatto ciò si è passato all'esportazione del database dal sito Altervista.

## 7.2 Esportazione del database dal sito Altervista

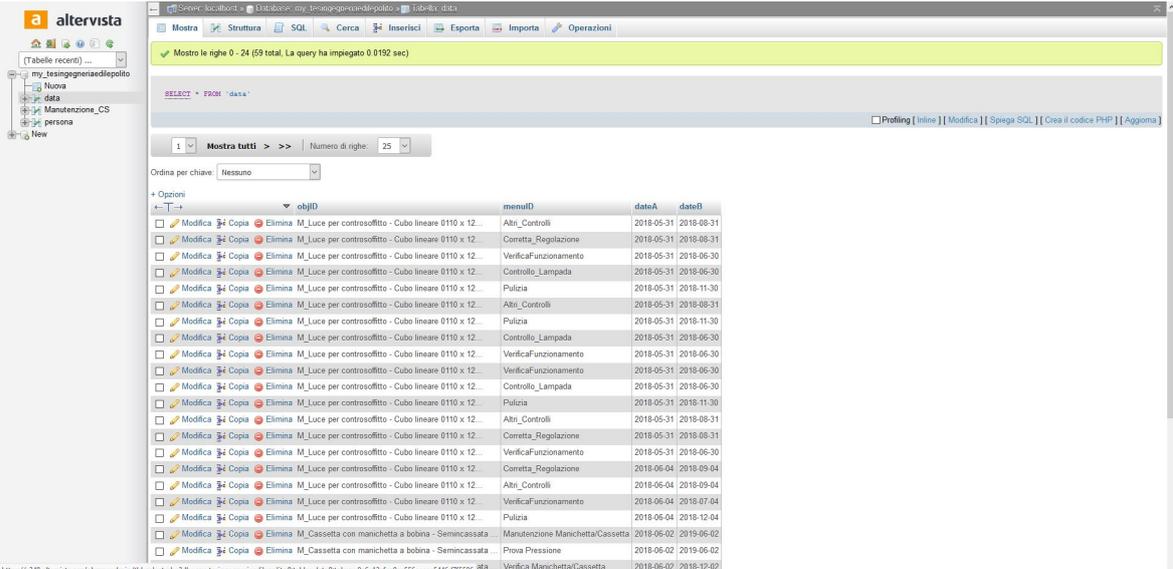
Per accedere alla pagina di gestione del sito, ricercando tramite un browser il sito di Altervista, è possibile accedere inserendo le credenziali fornite da Altervista stesso.

Una volta effettuato l'accesso è possibile visualizzare e accedere a tutte le informazioni riguardanti il database e gli script che inviano e ricevono informazioni dall'applicazione.

Per accedere a questi ultimi è possibile farlo entrando nella sezione denominata "Applicazioni" e successivamente cliccando su "Gestione File".

Mentre per accedere al database dalla bacheca basta selezionare la voce "Accedi a PhpMyAdmin" presente nella sezione "Risorse".

Da questa sezione è possibile visualizzare tutti i database, con le relative tabelle annesse, presenti all'interno del sito. In questo caso vi è un solo database denominato "my\_tesingegneriaedilepolito", il cui nome è assegnato di default. Al suo interno vi è una tabella denominata "data", all'interno della quale vi sono presenti tutti i dati del database attualmente salvati.



	id	memID	dataA	dataB
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Altri_Controlli	2018-05-31 2018-08-31
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Corretta_Regolazione	2018-05-31 2018-08-31
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	VerificaFunzionamento	2018-05-31 2018-06-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Controllo_Lampada	2018-05-31 2018-06-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Pulizia	2018-05-31 2018-11-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Altri_Controlli	2018-05-31 2018-08-31
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Pulizia	2018-05-31 2018-11-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Controllo_Lampada	2018-05-31 2018-06-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	VerificaFunzionamento	2018-05-31 2018-06-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	VerificaFunzionamento	2018-05-31 2018-06-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Controllo_Lampada	2018-05-31 2018-06-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Pulizia	2018-05-31 2018-11-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Altri_Controlli	2018-05-31 2018-08-31
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Corretta_Regolazione	2018-05-31 2018-08-31
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	VerificaFunzionamento	2018-05-31 2018-06-30
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Corretta_Regolazione	2018-06-04 2018-09-04
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Altri_Controlli	2018-06-04 2018-09-04
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	VerificaFunzionamento	2018-06-04 2018-07-04
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Luce per controsfitto - Cubo lineare 0110 x 12...	Pulizia	2018-06-04 2018-12-04
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Cassetta con manichetta a bobina - Semincassata	Manutenzione Manichetta/Cassetta	2018-06-02 2019-06-02
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Cassetta con manichetta a bobina - Semincassata	Prova Pressione	2018-06-02 2019-06-02
<input type="checkbox"/>	Modifica <input type="checkbox"/> Copia <input type="checkbox"/> Elimina	M_Cassetta con manichetta a bobina - Semincassata	Verifica Manichetta/Cassetta	2018-06-02 2018-12-02

Figura 98: Elementi attualmente presenti nel database

Tramite questa schermata è possibile accedere a numerose informazioni, tra le quali la struttura del database. Per l'esportazione, basta selezionare la voce "Esporta".

In questa schermata è possibile decidere i vari parametri dell'esportazione.

Le impostazioni scelte sono le seguenti:

**Output:**

Salva l'output in un file

Modello per nomi dei file:   usa questo per esportazioni future

Set di caratteri del file:

Compressione:

Salva l'output come testo

**Formato:**

**Opzioni specifiche al formato:**

Campi terminati con:

Campi limitati da:

Campi prefissati con:

Linee terminate con:

Sostituisci NULL con:

Rimuovi i caratteri di ritorno a capo/avanzamento riga (CR/LF) dall'interno dei campi

Metti i nomi dei campi nel primo rigo

Figura 99: Impostazioni per l'esportazione del database in formato CSV

### 7.3 Impostazione del DB Link in Revit

Una volta esportato il database da Altermista si è proceduto con la creazione del DB Link in Revit, utile ad esportare anche da Revit un database. Questa operazione è fondamentale in quanto, come si vedrà in seguito, questo database verrà implementato con le informazioni ottenute da quello di Altermista e successivamente reimportato.

Per fare ciò basta andare nella scheda Moduli aggiuntivi e selezionare DB Link.

Tra le varie alternative di esportazione è stato scelto quello denominato OCDB. Cliccando su “New Connection” e successivamente su “Export” si avvia la procedura di esportazione. Cliccando su “Nuovo” è possibile scegliere il driver per l’origine dati. Si sottolinea che se sul computer non fosse installato il pacchetto office, l’unica scelta sarebbe SQL Server. In questo caso invece è stato scelto “Microsoft Access Driver”.

Una volta creato il driver, il programma chiede di selezionare il file di Access a cui collegarsi. Per fare ciò è stato creato un file vuoto con Access e dalla schermata appena comparsa cliccando su “Select” è stato selezionato all’interno del suo percorso.

Una volta creato il DSN si verrà riportati alla schermata di partenza e questa volta invece di cliccare su Nuovo, basterà andare a selezionare il DSN dal percorso in cui era stato salvato. Fatto ciò, verrà avviata la procedura di esportazione.

Lighting fixtures	Id	IDTipo	Fasedicreaz	Fasedidemc	Variantidip	IDInterruto	Quadro	Numerodici	Datieletrici	Coefficienti	Commenti	Idhost	Livello	Contrassegr	ID_Element	Altri Control	Controllo La	Corretti	
828516	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						86	828516	Manutenzione	Manutenzione	Manute
828642	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						88	828642	Manutenzione	Manutenzione	Manute
828651	828604	86961					QE.S.P2.03 (O) 5		230 V/1-40 VA						89	828651	Manutenzione	Manutenzione	Manute
828659	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						90	828659	Manutenzione	Manutenzione	Manute
828669	828604	86961					QE.S.P2.03 (O) 1		230 V/1-40 VA						91	828669	Manutenzione	Manutenzione	Manute
828677	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						92	828677	Manutenzione	Manutenzione	Manute
828685	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						93	828685	Manutenzione	Manutenzione	Manute
828703	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						94	828703	Manutenzione	Manutenzione	Manute
828705	828604	86961					QE.S.P2.03 (O) 1		230 V/1-40 VA						95	828705	Manutenzione	Manutenzione	Manute
828706	828604	86961					QE.S.P2.03 (O) 5		230 V/1-40 VA						96	828706	Manutenzione	Manutenzione	Manute
828707	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						97	828707	Manutenzione	Manutenzione	Manute
828708	828604	86961					QE.S.P2.03 (O) 1		230 V/1-40 VA						98	828708	Manutenzione	Manutenzione	Manute
828709	828604	86961					QE.S.P2.03 (O) 5		230 V/1-40 VA						99	828709	Manutenzione	Manutenzione	Manute
828710	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						100	828710	Manutenzione	Manutenzione	Manute
828718	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						101	828718	Manutenzione	Manutenzione	Manute
828719	828604	86961					QE.S.P2.03 (O) 1		230 V/1-40 VA						102	828719	Manutenzione	Manutenzione	Manute
828720	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						103	828720	Manutenzione	Manutenzione	Manute
828721	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						104	828721	Manutenzione	Manutenzione	Manute
828722	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						105	828722	Manutenzione	Manutenzione	Manute
828723	828604	86961					QE.S.P2.03 (O) 5		230 V/1-40 VA						106	828723	Manutenzione	Manutenzione	Manute
828724	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						107	828724	Manutenzione	Manutenzione	Manute
828732	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						108	828732	Manutenzione	Manutenzione	Manute
828733	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						109	828733	Manutenzione	Manutenzione	Manute
828734	828604	86961					QE.S.P2.03 (O) 5		230 V/1-40 VA						110	828734	Manutenzione	Manutenzione	Manute
828735	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						111	828735	Manutenzione	Manutenzione	Manute
828736	828604	86961					QE.S.P2.03 (O) 1		230 V/1-40 VA						112	828736	Manutenzione	Manutenzione	Manute
828737	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						113	828737	Manutenzione	Manutenzione	Manute
828738	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						114	828738	Manutenzione	Manutenzione	Manute
828744	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						115	828744	Manutenzione	Manutenzione	Manute
828745	828604	86961					QE.S.P2.03 (O) 1		230 V/1-40 VA						116	828745	Manutenzione	Manutenzione	Manute
828746	828604	86961					QE.S.P2.03 (O) 5		230 V/1-40 VA						117	828746	Manutenzione	Manutenzione	Manute
828747	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						118	828747	Manutenzione	Manutenzione	Manute
828748	828604	86961					QE.S.P2.03 (O) 1		230 V/1-40 VA						119	828748	Manutenzione	Manutenzione	Manute
828749	828604	86961					QE.S.P2.03 (O) 5		230 V/1-40 VA						120	828749	Manutenzione	Manutenzione	Manute
828750	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						121	828750	Manutenzione	Manutenzione	Manute
828758	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						122	828758	Manutenzione	Manutenzione	Manute
828759	828604	86961					QE.S.P2.03 (O) 1		230 V/1-40 VA						123	828759	Manutenzione	Manutenzione	Manute
828760	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						124	828760	Manutenzione	Manutenzione	Manute
828761	828604	86961					QE.S.P2.03 (O) 3		230 V/1-40 VA						125	828761	Manutenzione	Manutenzione	Manute
828762	828604	86961					QE.S.P2.03 (O) 2		230 V/1-40 VA						126	828762	Manutenzione	Manutenzione	Manute

Figura 100: Database di Revit visualizzato in Access

Aperto il precedente file di Access appena creato, è adesso possibile visualizzare i medesimi dati visualizzabili in Revit. Nelle colonne relative alla manutenzione è stato inserito il testo “Manutenzione mai avvenuta!”. Nella colonna “ID\_Elemento” invece è stato fatto combaciare il medesimo numero presente nella colonna “Id”. Questo passaggio è stato fatto per permettere di visualizzare l’id nelle proprietà istanza di ogni singolo elemento senza dover andare ogni volta a selezionare l’apposito comando e per poterlo visualizzare all’interno dell’abaco. Questi campi verranno successivamente aggiornati con i dati contenuti nel database di Altervista.

## 7.4 Unione dei database e creazione dell'abaco delle manutenzioni

Una volta esportati i database da Revit e da Altermvista, si è proceduto all'importazione dei dati da quest'ultimo in quello generato con il DB Link.

A titolo esemplificativo vengono riportati i dati riguardanti solo alcuni elementi.

ID	IDtipo	Fased	Quadro	N	Contrassegr	ID_Element	Altri Controlli	Controllo Lampada	Corretta Regolazione
829472	826669	88961	QE.S.P2.04 (O,1,8			217	829472	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829480	826669	88961	QE.S.P2.04 (O,1,8			218	829480	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829488	826669	88961	QE.S.P2.04 (O,1,10			219	829488	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829496	826669	88961	QE.S.P2.04 (O,1,1			220	829496	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829504	826669	88961	QE.S.P2.04 (O,1,8			221	829504	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829512	826669	88961	QE.S.P2.04 (O,1,8			222	829512	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829520	826669	88961	QE.S.P2.04 (O,1,1			223	829520	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829528	826669	88961	QE.S.P2.04 (O,1,9			224	829528	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829536	826669	88961	QE.S.P2.04 (O,1,8			225	829536	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829544	826669	88961	QE.S.P2.04 (O,1,8			226	829544	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829552	826669	88961	QE.S.P2.04 (O,1,1			227	829552	Data Manutenzione: 31/05/2018 Deadline: 31/08/2018	Data Manutenzione: 31/05/2018 Deadline: 31/08/2018
829560	826669	88961	QE.S.P2.04 (O,1,8			228	829560	Data Manutenzione: 31/05/2018 Deadline: 31/08/2018	Data Manutenzione: 31/05/2018 Deadline: 31/08/2018
829568	826669	88961	QE.S.P2.04 (O,1,10			229	829568	Data Manutenzione: 31/05/2018 Deadline: 31/08/2018	Data Manutenzione: 31/05/2018 Deadline: 31/08/2018
829576	826669	88961	QE.S.P2.04 (O,1,8			230	829576	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829584	826669	88961	QE.S.P2.04 (O,1,1			231	829584	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829592	826669	88961	QE.S.P2.04 (O,1,8			232	829592	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829600	826669	88961	QE.S.P2.04 (O,1,8			233	829600	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829608	826669	88961	QE.S.P2.04 (O,1,5			234	829608	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829616	826669	88961	QE.S.P2.04 (O,1,7			235	829616	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829624	826669	88961	QE.S.P2.04 (O,1,5			236	829624	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829632	826669	88961	QE.S.P2.04 (O,1,5			237	829632	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829640	826669	88961	QE.S.P2.04 (O,1,10			238	829640	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829648	826669	88961	QE.S.P2.04 (O,1,7			239	829648	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829656	826669	88961	QE.S.P2.04 (O,1,5			240	829656	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829664	826669	88961	QE.S.P2.04 (O,1,5			241	829664	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829672	826669	88961	QE.S.P2.04 (O,1,7			242	829672	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829680	826669	88961	QE.S.P2.04 (O,1,5			243	829680	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829688	826669	88961	QE.S.P2.04 (O,1,5			244	829688	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829696	826669	88961	QE.S.P2.04 (O,1,5			245	829696	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829704	826669	88961	QE.S.P2.04 (O,1,7			246	829704	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829712	826669	88961	QE.S.P2.04 (O,1,5			247	829712	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829720	826669	88961	QE.S.P2.04 (O,1,10			248	829720	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829728	826669	88961	QE.S.P2.04 (O,1,5			249	829728	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829736	826669	88961	QE.S.P2.04 (O,1,7			250	829736	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829744	826669	88961	QE.S.P2.04 (O,1,5			251	829744	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829752	826669	88961	QE.S.P2.04 (O,1,5			252	829752	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829760	826669	88961	QE.S.P2.04 (O,1,5			253	829760	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829768	826669	88961	QE.S.P2.04 (O,1,5			254	829768	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829776	826669	88961	QE.S.P2.04 (O,1,9			255	829776	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!
829784	826669	88961	QE.S.P2.04 (O,1,5			256	829784	Manutenzione Mai Eseguita!!	Manutenzione Mai Eseguita!!

Figura 101: Dati importati nel database collegato con Revit

Una volta aggiornato questo database si è proceduto alla reimportazione del materiale aggiornato in Revit.

Accedendo alla medesima scheda per l'esportazione, selezionando il collegamento nel quale è presente il DSN e cliccando su "Edit and Import", verrà avviata la procedura di importazione dei dati.

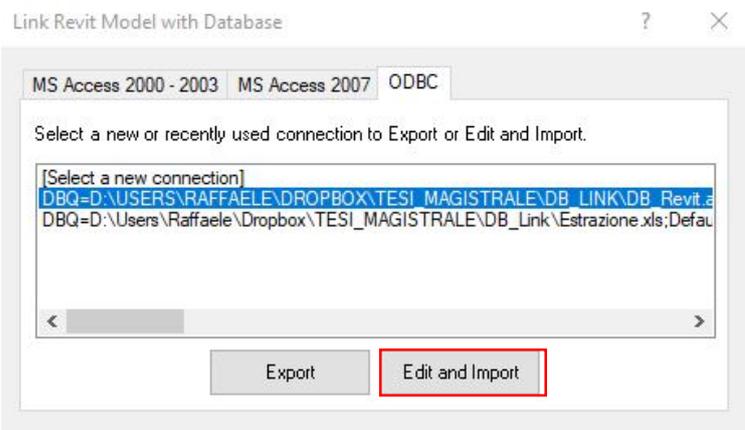


Figura 102: Importazione del database aggiornato

Una volta finita l'importazione si otterrà un output nel quale viene riepilogato i dati che sono stati importati e se ci sono stati degli errori. I vari output vengono colorati in base al tipo di esito ottenuto.

Appearance	Meaning
Blue	Updated successfully
Red	Update failed
Fuchsia	The corresponding parameter is ReadOnly
DarkOrchid	The corresponding parameter does not exist
Violet	Value is empty
Gray	Value has no change
Yellow	Exception
Orange	UnKnown
Rows that failed to add	Rows that failed to add
Rows that added successfully	Rows that added successfully

Figura 104: Legenda degli output

ID	Modello	Parametro	Valore	Stato	Errore	Colore	Descrizione	Data	Data	Data	Data	Data
829552	826669	86961	(null)	(null)	(null)	Blue	QE S.P2.04 (O.P15.20), 230 V/400 V, Tre Fasi, 4 Carri, Raccordo a Y	31/05/2018	31/05/2018	31/05/2018	31/05/2018	31/05/2018
829560	826669	86961	(null)	(null)	(null)	Blue	QE S.P2.04 (O.P15.20), 230 V/400 V, Tre Fasi, 4 Carri, Raccordo a Y	31/05/2018	31/05/2018	31/05/2018	31/05/2018	31/05/2018
829568	826669	86961	(null)	(null)	(null)	Blue	QE S.P2.04 (O.P15.20), 230 V/400 V, Tre Fasi, 4 Carri, Raccordo a Y	31/05/2018	31/05/2018	31/05/2018	31/05/2018	31/05/2018

Figura 103: Esito dell'importazione.

Una volta importati tutti i dati, si è proceduto alla creazione dell'abaco delle manutenzioni dei dispositivi di illuminazione. Riutilizzando l'abaco creato per i dispositivi di illuminazione, sono stati aggiunti i parametri condivisi precedentemente mostrati. Il risultato ottenuto è il seguente.

<Abaco della Manutenzione dei Dispositivi di Illuminazione>							
A	B	C	D	E	F	G	H
Tipo	Contrassegno	ID_Elemento	Controllo Lampada	Corretta Regolazione	Pulizia	Verifica Funzionamento	Altri Controlli
M_Luce per controsoffitto - Cubo lineare							
0110 x 1200 mm (2) 227	829552		Data Manutenzione: 31/05/2018				
0110 x 1200 mm (2) 228	829560		Data Manutenzione: 31/05/2018				
0110 x 1200 mm (1) 257	829760		Data Manutenzione: 31/05/2018				
0110 x 1200 mm (2) 191	829082		Manutenzione Mai Eseguita!				
0110 x 1200 mm (2) 192	829303		Manutenzione Mai Eseguita!				
0110 x 1200 mm (2) 193	829314		Manutenzione Mai Eseguita!				
0110 x 1200 mm (2) 194	828322		Manutenzione Mai Eseguita!				
Data Manutenzione: 31/05/2018	Deadline: 30/06/2018	Data Manutenzione: 31/05/2018	Deadline: 31/08/2018	Data Manutenzione: 31/05/2018	Deadline: 30/11/2018	Data Manutenzione: 31/05/2018	Deadline: 31/08/2018
Data Manutenzione: 31/05/2018	Deadline: 30/06/2018	Data Manutenzione: 31/05/2018	Deadline: 31/08/2018	Data Manutenzione: 31/05/2018	Deadline: 30/11/2018	Data Manutenzione: 31/05/2018	Deadline: 31/08/2018
Data Manutenzione: 31/05/2018	Deadline: 30/06/2018	Manutenzione Mai Eseguita!		Data Manutenzione: 31/05/2018	Deadline: 30/11/2018	Data Manutenzione: 31/05/2018	Deadline: 31/08/2018
Data Manutenzione: 31/05/2018	Deadline: 30/06/2018	Manutenzione Mai Eseguita!		Data Manutenzione: 31/05/2018	Deadline: 30/11/2018	Data Manutenzione: 31/05/2018	Deadline: 31/08/2018

Essendo gli elementi dell'abaco direttamente collegati con gli elementi presenti nel modello, andando a selezionare i singoli elementi nel modello è possibile visualizzare le medesime informazioni tramite il menu delle proprietà istanza.

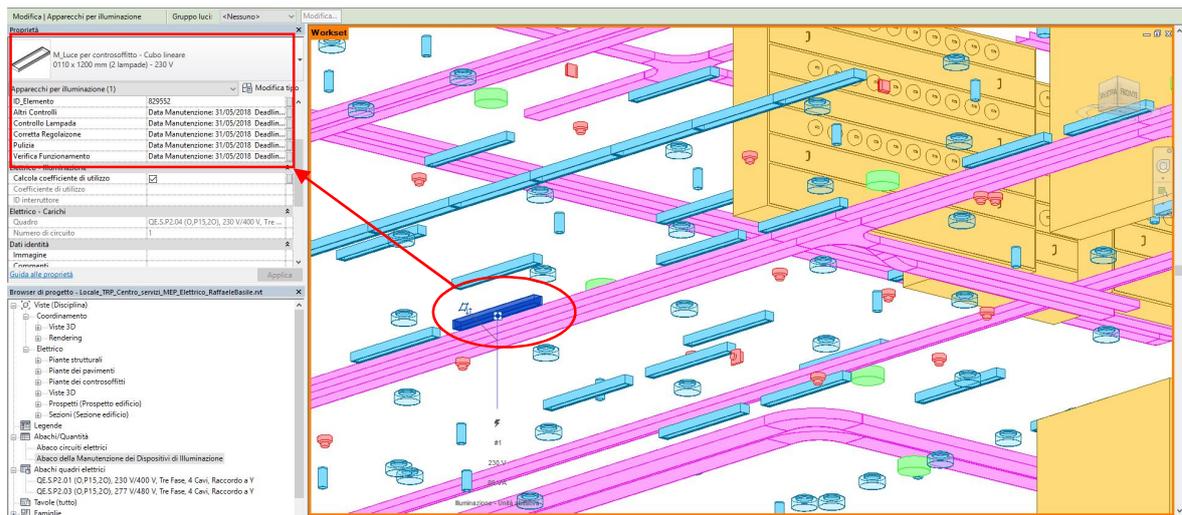


Figura 105: Visualizzazione delle proprietà istanza dalla vista 3D

## 8 CONCLUSIONI E SVILUPPI FUTURI

### 8.1 Conclusioni e sviluppi futuri sul processo di interoperabilità e datasharing

Si può affermare che il processo di interoperabilità è risultato essere efficiente.

Durante l'esportazione in Advance Steel e durante la successiva reimportazione in Revit, i programmi sono riusciti brillantemente a riconoscere le modifiche fatte rispetto alla precedente esportazione. Inoltre anche la conversione dei materiali e delle sezioni è risultata essere molto efficiente. Infatti quando non venivano riconosciute determinate sezioni, è stato possibile andare a selezionare manualmente il tipo di sezione in cui convertire quella proveniente dall'altro software.

Per quanto riguarda l'interoperabilità tra Revit e Unity3D, anch'essa non ha generato alcun tipo di problema a livello geometrico. Per quanto riguarda l'impianto elettrico, ad esempio, sono state perse tutte le informazioni di tipo alfanumerico quali ad esempio le informazioni sui circuiti a cui gli elementi erano collegati e quali circuiti erano collegati ai vari quadri. Questo comunque non è risultato essere un problema in quanto l'attenzione si è concentrata sulle operazioni di manutenzione da effettuare, informazioni che originariamente non erano presenti in Revit.

Per quanto riguarda il datasharing si può affermare che questo funzioni efficacemente. I dati acquisiti attraverso l'applicazione sono stati importati all'interno del modello BIM in maniera piuttosto semplice. Sicuramente c'è da implementare una regola per i database che permetta di importare i dati in maniera automatica in quanto allo stato attuale ciò è possibile farlo solo manualmente.

## 8.2 Conclusioni e sviluppi futuri del modello strutturale

Si può affermare che certamente la modellazione con il plug-in Advance Steel abbia decisamente agevolato la parte di modellazione in quanto ha consentito di evitare la parametrizzazione delle famiglie dei giunti in acciaio; si è rivelato essere uno strumento potente ma ancora acerbo e che necessita di ulteriori sviluppi.

Innanzitutto non si può ritenere accettabile la limitazione nell'estrapolazione dei dati per la creazione degli abachi, inoltre l'impostazione del progetto risulta essere troppo macchinosa in quanto prima di iniziare a modellare i giunti bisogna prima importare le famiglie certificate da librerie di altri stati.

Infine risulta ancora impossibile la realizzazione di giunti tra elementi di materiali diversi. Ad esempio non è stato possibile creare le famiglie di giunti che andassero a collegare il telaio strutturale in acciaio con i setti, in quanto per realizzare quest'ultimi sono stati usati dei muri composti di calcestruzzo che però Advance Steel non riconosce come elementi strutturali.

Sicuramente un ulteriore studio che si potrà effettuare su questo modello strutturale è quello di testare l'interoperabilità che c'è tra Revit e Advance Design e provare le funzionalità di questo software.

Questo programma sembra essere molto promettente in quanto permette di fare analisi FEM, analisi sismiche e, infine, di adattare i vari calcoli strutturali a tutte le normative di riferimento nel mondo in modo tale da poter essere utilizzabile non solo su suolo italiano.

Advance Design è di proprietà della stessa casa software che possedeva Advance Steel prima che fosse venduto alla Autodesk, dunque ci si aspetta un elevato grado di interoperabilità tra i due software.

## 8.3 Conclusioni e sviluppi futuri del modello MEP

La progettazione MEP non ha dato particolari problemi: è risultato essere un metodo di progettazione molto potente che sicuramente semplifica di molto la progettazione e che permette di prevenire errori quali ad esempio l'interferenza tra elementi di discipline diverse. L'unico problema riscontrato durante la modellazione è stato relativo alle quote delle passerelle elettriche. Poiché si disponeva solo di 3 sezioni per tutto l'edificio dalle quali si riusciva

ad evincere la loro quota esatta, le altre quote sono state ipotizzate facendo passare le canalette attraverso i fori presenti nelle travi assumendo le distanze dai bordi dei fori simili a quelle presenti nelle sezioni.

Sicuramente lo sviluppo del modello MEP consisterà nell'implementazione degli altri sistemi non ancora modellati.

## 8.4 Conclusioni e sviluppi futuri dell'applicazione per la manutenzione in Realtà Aumentata

L'applicazione creata è risultata essere un mezzo molto utile oltre che per tenere traccia delle manutenzioni da eseguire, anche per valutare se in fase di cantiere c'è corrispondenza tra ciò che è stato definito in fase di progetto e ciò che effettivamente viene realizzato.

Ad esempio durante la fase di sopralluogo è stata riscontrata un'incongruenza. Gli sprinkler infatti sono risultati essere spostati rispetto alla loro posizione prevista di circa 50cm.

La creazione di questa applicazione è stata molto difficile e ha occupato quasi la maggior parte del tempo dedicato alla creazione del presente lavoro. La difficoltà principale è stata certamente la programmazione dato che nel corso di studi di ingegneria edile non vi è un corso di programmazione ad oggetti e Unity si basa proprio su questo tipo di programmazione. Certamente è stato di aiuto il corso del primo anno di informatica il quale ha messo le basi per potersi avvicinare a questo tipo di programmazione.

Tanto è stato difficile e tanto è stato soddisfacente realizzare questa applicazione. Si è avuta la possibilità di testare quali sono le nuove tecnologie e di affacciarsi al mondo della realtà aumentata e virtuale le quali sicuramente in futuro giocheranno un ruolo importante nel mondo della progettazione.

Di sicuro questa applicazione attualmente è a livello embrionale.

Uno degli sviluppi sarà sicuramente l'implementazione degli altri modelli della sede unica della regione Piemonte in modo tale da poter tenere sotto controllo tutte le informazioni riguardanti la manutenzione da un'unica applicazione.

Sicuramente una miglioria che si potrebbe apportare è quella di rendere l'applicazione un ibrido tra realtà aumentata per ambiente e per marker. Attualmente, come visto in precedenza, per collocare il modello bisogna far coincidere l'origine settata all'interno di Unity3D con l'analogo punto nella realtà. Questo metodo si porta dietro ovvi errori di precisione che

possono essere compensati con un allineamento tra edificio reale e modello virtuale tramite, ad esempio, il collocamento di codici a barre o QR come immagini target in punti noti dell'edificio.

Inoltre, un upgrade sarà sicuramente la creazione di un sistema di notifiche push che avvisino i manutentori qualche giorno prima della scadenza di una manutenzione dove bisognerà andare a eseguire le varie operazioni.

Quest'ultimo upgrade non è stato fatto in questo lavoro in quanto non si dispone di un server adatto a questo scopo né delle competenze che risultano essere decisamente più specifiche e di settore rispetto a quelle richieste per la realizzazione di questa applicazione allo stato attuale.

Infine un ultimo aspetto da sviluppare è quello riguardante la connessione dei due database. Attualmente i due database non si scambiano informazioni in modo automatico in quanto presentano una struttura diversa tra di loro. Una miglioria che si potrebbe fare in futuro potrebbe essere quella di creare una regola in Access che permetta in automatico la ricerca degli ID riguardanti il medesimo elemento e che ricopi in automatico le date del database di Altvista nelle colonne giuste di quello di Revit.

## 9 RINGRAZIAMENTI

Anche per questa seconda laurea, come per quella triennale, i primi ringraziamenti vanno alla mia famiglia. Senza il loro sostegno morale ed economico e senza i loro consigli di sicuro non sarei arrivato al raggiungimento di questo traguardo così importante e in questi tempi.

Un ringraziamento va alla professoressa Anna Osello, a Matteo Del Giudice e a Giulio Di Vico per avermi accolto in dipartimento, che si è rivelato essere una seconda casa, per avermi seguito durante questo lungo anno sia durante il tirocinio sia durante la stesura di questa tesi. Grazie a loro ho arricchito notevolmente il mio bagaglio culturale e ampliato le mie conoscenze; tramite questa esperienza ho avuto la possibilità di mettere mano su tecnologie con le quali difficilmente avrei lavorato se non avessi lavorato presso il dipartimento.

Ringrazio la Regione Piemonte per avermi dato la possibilità di lavorare su un progetto di una così grande importanza e l'ingegnere Borgogno per la disponibilità sempre mostrata soprattutto durante i sopralluoghi in cantiere.

Un ringraziamento va a Federico Cucinella grazie al quale è stato possibile realizzare il database in Cloud per il salvataggio dei dati.

Infine un ringraziamento va a tutti i miei amici, in particolare Mirko Blasi, Francesco Carbotti e Giuseppe Africola per il costante sostegno morale. A Giuseppe devo anche la riuscita di tutto ciò che è contenuto nel presente elaborato: senza il suo costante supporto informatico ad ogni problema avuto durante il corso di quest'anno con i computer, difficilmente avrei portato a termine questo lavoro in questi tempi.

## 10 BIBLIOGRAFIA

La sfida del BIM – Chiara C. Rizzarda – Gabriele Gallo

Unity Virtual Reality Projects – Jonathan Linowes

La realidad aumentada como medio de visualización del modelo BIM en la construcción –  
Gerardo Miguel Chavarri De Lemos

BIM per As-Built Caso studio della “Torre della Regione” a Torino” – Giulio Di Vico

Maturità digitale BIM con gestione dati automatizzata per l’Allianz Stadium – Lorenzo Iaconinoto

## 11 SITOGRAFIA

<http://www.fuksas.it/it/Progetti/Nuova-Sede-Regione-Piemonte-Turin/>

[Ultimo accesso: 25/02/2018]

<http://www.regione.piemonte.it/sedeunica/>

[Ultimo accesso: 27/02/2018]

<https://it.graitec.com/autodesk-advance-steel/>

[Ultimo accesso: 15/04/2018]

<https://knowledge.autodesk.com/it/support/revit-products/troubleshooting/caas/sfdarticles/sfdarticles/ITA/How-to-use-certified-families-for-a-number-of-language-packs-for-Steel-Connection-tool-in-Revit-2017.html>

[Ultimo accesso: 20/04/2018]

<https://knowledge.autodesk.com/it/support/revit-products/learn-explore/caas/Cloud-Help/cloudhelp/2017/ITA/Revit-AddIns/files/GUID-6244E741-5D14-4DAD-AE25-5069F71B69F3-htm.html>

[Ultimo accesso: 30/04/2018]

<http://help.autodesk.com/view/RVT/2016/ITA/?guid=GUID-8BB35A3A-475A-4FE0-99C0-B01054EE0F2A>

[Ultimo accesso: 05/05/2018]

<https://answers.unity.com/index.html>

[Ultimo accesso: 01/06/2018]

<https://developer.apple.com/arkit/>

[Ultimo accesso: 01/06/2018]

<https://www.vuforia.com/>

[Ultimo accesso: 01/06/2018]

<https://www.facebook.com/groups/356466471533309/>

[Ultimo accesso: 06/06/2018]

<https://www.youtube.com/user/Brackeys>

[Ultimo accesso: 15/06/2018]

[https://www.youtube.com/channel/UCZBK18\\_FerdLIQ3qTp3GuJw](https://www.youtube.com/channel/UCZBK18_FerdLIQ3qTp3GuJw)

[Ultimo accesso: 26/06/2018]

<http://tesingegneriapolito.altervista.org>

[Ultimo accesso: 01/07/2018]

## 12 ALLEGATI

### 12.1 Rapporto giunto

Giunto angolare sghembo

Standard: EC3

Giunto angolare sghembo Descrizione

Verifica dei bulloni sul profilo principale

Condizioni

Verifica a taglio dei bulloni

Verifica a rifollamento bulloni

Verifica a taglio dell'angolare sul profilo principale

Verifica a snervamento per taglio

Verifica a taglio ultimo

Verifica a rottura in blocco

Verifica dei bulloni sul profilo secondario

Condizioni

Verifica a taglio dei bulloni

Verifica a rifollamento bulloni

Verifica a taglio dell'angolare sul profilo secondario

Verifica a snervamento per taglio

Verifica a taglio ultimo

Verifica a rottura in blocco

Verifica a taglio del profilo secondario

Verifica a snervamento per taglio

Verifica a taglio ultimo

Verifica a rottura in blocco

Verifica della sezione di taglio

Verifica della sezione

Conclusione

Il giunto non resiste ai carichi applicati

Giunto angolare sghembo Descrizione

#### Elementi collegati

Elemento	Profili	Altezza	Larghezza	Spessore anima	Spessore ala	Raggio raccordo	Materiale	ID
Profilo principale	HEA100	96mm	100mm	5mm	8mm	12mm	S355JR	2
Trave secondaria	HEA100	96mm	100mm	5mm	8mm	12mm	S355JR	3

#### Forze di progetto

Nome caso	M	N	V
SimpleMaxTorsor	0kNm	20kN	10kN

#### Proprietà Bulloni

Connettere	Tipo	Diametro	Nr. Bulloni	Classe	Assembly	Diametro del foro	Area
Main beam - Clip Angle	UNI 5712	16mm	6	10.9	Mu2S	2mm	0m <sup>2</sup>
Secondary beam - Clip Angle	UNI 5712	16mm	3	10.9	Mu2S	2mm	0m <sup>2</sup>

#### Verifica dei bulloni sul profilo principale

##### Condizioni

#### Dettagli del giunto

Distanza min/max dei bulloni dal bordo (nella direzione del carico)

$$1.2 * d_0 \leq e_1 \leq (4 * t + 40\text{mm})$$

$$21.6\text{mm} \leq 25\text{mm} \leq 60\text{mm}$$

OK

Distanza min/max dei bulloni dal bordo (perpendicolare alla direzione del carico)

$$1.2 * d_0 \leq e_2 \leq (4 * t + 40\text{mm})$$

$$21.6\text{mm} \leq 27.5\text{mm} \leq 60\text{mm}$$

OK

Distanza intermedia bulloni min/max (nella direzione del carico)

$$2.2 * d_0 \leq p_1 \leq \min(14 * t; 200\text{mm})$$

$$39.6\text{mm} \leq 50\text{mm} \leq 70\text{mm}$$

OK

Verifica a taglio dei bulloni

$$F_{v,Ed} \leq F_{v,Rd}$$

$$\text{SQRT}((F_{v,Ed}, N_x + F_{v,Ed}, M_{xi})^2 + (F_{v,Ed}, V_y + F_{v,Ed}, M_{yi})^2) \leq n_s * \alpha_v * f_{ub} * A_s / \gamma_{Mb}$$

$$\text{SQRT}((0\text{kN} + 0\text{kN})^2 + (0\text{kN} + 0\text{kN})^2) \leq 2 * 0.5 * 1000\text{N/mm}^2 * 0\text{m}^2 / 1.25$$

$$0\text{kN} \leq 125.6\text{kN}$$

0 %

OK

Verifica a rifollamento bulloni

Resistenza dei bulloni sull'angolare

$$F_{v,Ed} \leq F_{b,Rd}$$

$$F_{v,Ed} \leq 1 * k_1 * \alpha_b * f_u * d * \Sigma(t_i) / \gamma_{Mb}$$

$$0kN \leq 1 * 2.5 * 0.46 * 510N/mm^2 * 16mm * 20mm / 1.25$$

$$0kN \leq 151.1kN$$

0 % OK

Resistenza dei bulloni sul profilo principale

$$F_{v,Ed} \leq F_{b,Rd}$$

$$F_{v,Ed} \leq 1 * k_1 * \alpha_b * f_u * d * \Sigma(t_i) / \gamma_{Mb}$$

$$0kN \leq 1 * 2.5 * -2.07 * 510N/mm^2 * 16mm * 5mm / 1.25$$

$$0kN \leq -169.2kN$$

0 %

Non riuscito

Verifica a taglio dell'angolare sul profilo principale

Verifica a snervamento per taglio

$$V_{Ed} \leq V_{pl,Rd}$$

$$V_{Ed} \leq n_{Obj} * f_y * A_v / (\gamma_{M0} * \text{SQRT}(3))$$

$$10kN \leq 2 * 355N/mm^2 * 0m^2 / (1 * 1.73)$$

$$10kN \leq 614.9kN$$

1.63 %

OK

Verifica a taglio ultimo

$$V_{Ed} \leq V_{u,Rd}$$

$$V_{Ed} \leq n_{Obj} * 0.9 * f_u * A_{v,Net} / (\gamma_{M2} * \text{SQRT}(3)) \quad 10kN \leq 2 * 0.9 * 510N/mm^2 * 0m^2 / (1.25 * 1.73)$$

$$10kN \leq 407kN$$

2.46 %

OK

Verifica a rottura in blocco

$$V_{Ed} \leq V_{eff,2,Rd}$$

$$V_{Ed} \leq n_{Obj} * (0.5 * f_u * A_{nt} / \gamma_{M2} + f_y * A_{nv} / (\gamma_{M0} * \text{SQRT}(3))) \quad 10kN \leq 2 * (0.5 * 510N/mm^2 * 0m^2 / 1.25 + 355N/mm^2 * 0m^2 / (1 * 1.73))$$

$$10kN \leq 403.4kN$$

2.48 %

OK

## Verifica dei bulloni sul profilo secondario

---

### Condizioni

Distanza min/max dei bulloni dal bordo (nella direzione del carico)

$$1.2 * d_0 \leq e_1 \leq (4 * t + 40\text{mm})$$

$$21.6\text{mm} \leq 25\text{mm} \leq 60\text{mm}$$

OK

Distanza min/max dei bulloni dal bordo (perpendicolare alla direzione del carico)

$$1.2 * d_0 \leq e_2 \leq (4 * t + 40\text{mm})$$

$$21.6\text{mm} \leq 10\text{mm} \leq 60\text{mm} \text{ Non riuscito}$$

Distanza intermedia bulloni min/max (nella direzione del carico)

$$2.2 * d_0 \leq p_1 \leq \min(14 * t; 200\text{mm})$$

$$39.6\text{mm} \leq 50\text{mm} \leq 70\text{mm}$$

OK

Verifica a taglio dei bulloni

$$F_{v,Ed} \leq F_{v,Rd}$$

$$\text{SQRT}((F_{v,Ed, N_x} + F_{v,Ed, M_{x_i}})^2 + (F_{v,Ed, V_y} + F_{v,Ed, M_{y_i}})^2) \leq n_s * \alpha_v * f_{ub} * A_s / \gamma_{Mb}$$

$$\text{SQRT}((6.7\text{kN} + 6.5\text{kN})^2 + (3.3\text{kN} + 0\text{kN})^2) \leq 2 * 0.5 * 1000\text{N/mm}^2 * 0\text{m}^2 / 1.25$$

$$13.6\text{kN} \leq 125.6\text{kN}$$

$$10.81 \%$$

OK

Verifica a rifollamento bulloni

Resistenza dei bulloni sull'angolare

$$F_{v,Ed} \leq F_{b,Rd}$$

$$F_{v,Ed} \leq 1 * k_1 * \alpha_b * f_u * d * \Sigma(t_i) / \gamma_{Mb}$$

$$13.6\text{kN} \leq 1 * 2.5 * 0.46 * 510\text{N/mm}^2 * 16\text{mm} * 20\text{mm} / 1.25$$

$$13.6\text{kN} \leq 151.1\text{kN}$$

$$8.99 \%$$

OK

Resistenza dei bulloni sul profilo secondario

$$F_{v,Ed} \leq F_{b,Rd}$$

$$F_{v,Ed} \leq 1 * k_1 * \alpha_b * f_u * d * \Sigma(t_i) / \gamma_{Mb}$$

$$13.6\text{kN} \leq 1 * 2.5 * -2.41 * 510\text{N/mm}^2 * 16\text{mm} * 5\text{mm} / 1.25$$

$$13.6\text{kN} \leq -196.4\text{kN}$$

6.91 %

Non riuscito

Verifica a taglio dell'angolare sul profilo secondario

---

Verifica a snervamento per taglio

$V_{Ed} \leq V_{pl,Rd}$

$V_{Ed} \leq n_{Obj} * f_y * A_v / (\gamma_{M0} * \text{SQRT}(3))$   $10\text{kN} \leq 2 * 355\text{N/mm}^2 * 0\text{m}^2 / (1 * 1.73)$

$10\text{kN} \leq 614.9\text{kN}$

1.63 %

OK

Verifica a taglio ultimo

$V_{Ed} \leq V_{u,Rd}$

$V_{Ed} \leq n_{Obj} * 0.9 * f_u * A_{v,Net} / (\gamma_{M2} * \text{SQRT}(3))$   $10\text{kN} \leq 2 * 0.9 * 510\text{N/mm}^2 * 0\text{m}^2 / (1.25 * 1.73)$

$10\text{kN} \leq 407\text{kN}$

2.46 %

OK

Verifica a rottura in blocco

$V_{Ed} \leq V_{eff,2,Rd}$

$V_{Ed} \leq n_{Obj} * (0.5 * f_u * A_{nt} / \gamma_{M2} + f_y * A_{nv} / (\gamma_{M0} * \text{SQRT}(3)))$   $10\text{kN} \leq 2 * (0.5 * 510\text{N/mm}^2 * 0\text{m}^2 / 1.25 + 355\text{N/mm}^2 * 0\text{m}^2 / (1 * 1.73))$

$10\text{kN} \leq 332\text{kN}$

3.01 %

OK

Verifica a taglio del profilo secondario

---

Verifica a snervamento per taglio

$V_{Ed} \leq V_{pl,Rd}$

$V_{Ed} \leq n_{Obj} * f_y * A_v / (\gamma_{M0} * \text{SQRT}(3))$   $10\text{kN} \leq 1 * 355\text{N/mm}^2 * 0\text{m}^2 / (1 * 1.73)$

$10\text{kN} \leq 153.7\text{kN}$

6.51 %

OK

Verifica a taglio ultimo

$V_{Ed} \leq V_{u,Rd}$

$$V_{Ed} \leq n_{Obj} * 0.9 * f_u * A_{v,Net} / (\gamma_{M2} * \text{SQRT}(3)) \quad 10\text{kN} \leq 1 * 0.9 * 510\text{N/mm}^2 * 0\text{m}^2 / (1.25 * 1.73)$$

$$10\text{kN} \leq 101.8\text{kN}$$

9.83 %

OK

Verifica a rottura in blocco

$$V_{Ed} \leq V_{eff,2,Rd}$$

$$V_{Ed} \leq n_{Obj} * (0.5 * f_u * A_{nt} / \gamma_{M2} + f_y * A_{nv} / (\gamma_{M0} * \text{SQRT}(3))) \quad 10\text{kN} \leq 1 * (0.5 * 510\text{N/mm}^2 * 0\text{m}^2 / 1.25 + 355\text{N/mm}^2 * 0\text{m}^2 / (1 * 1.73))$$

$$10\text{kN} \leq -29.9\text{kN}$$

33.4 %

Non riuscito

Verifica della sezione di taglio

---

Verifica della sezione

L'effetto della forza assiale è trascurato(ad esempio:  $N_{Ed} / N_{pl,Rd} \leq 0.25$ ;  $N_{Ed} \leq 0.5 * h_w * t_w * f_y / \gamma_{M0}$ )

Situazione:  $V_{Ed} \leq 0.5 * V_{pl,Rd}$

$$M_{Ed} \leq M_{v,Rd}$$

$$V_{Ed} * e \leq W_{pl} * f_y / \gamma_{M0}$$

$$10\text{kN} * 52.5\text{mm} \leq 4.5\text{cm}^3 * 355\text{N/mm}^2 / 1$$

$$0.5\text{kNm} \leq 1.6\text{kNm}$$

32.86 %

OK

Conclusione

---

Il giunto non resiste ai carichi applicati

## 12.2 Abaco delle connessioni strutturali

<Abaco delle connessioni strutturali>					
A	B	C	D	E	F
Famiglia	Tipo	Conteggio	Bolt Size	Number Of Bolts	Livello
Baseplate	Baseplate	54			
Clip angle - Skewed	IPE 140	157			
Clip angle - Skewed	IPE 240 - IPE 240	4			
Clip angle - Skewed	IPE 240 - IPE 300	1			
Clip angle - Skewed	IPE 270 - HEB 400	114			
Clip angle - Skewed	IPE 270 - IPE 270	122			
Clip angle - Skewed	IPE 270 - IPE 720	162			
Clip angle - Skewed	Ipe 300 - IPE 300 (S)	1			
Clip angle - Skewed	IPE 300 - IPE 400	4			
Clip angle - Skewed	IPE 400 - HEB 800	34			
Clip angle - Skewed	IPE 400 - IPE240	34			
Clip angle - Skewed	IPE 400 - IPE 270	14			
Clip angle - Skewed	IPE 400 - IPE 400	4			
Clip angle - Skewed	IPE 400 - IPE 720	28			
Front plate splice	Front plate splice	4			
Front plate splice	HEB 400 - HEB 400	75			
Front plate splice	HEB 400 - HEM 800	1			
Front plate splice	HEM 800 - HEM 800	6			
Front plate splice	IPE240 - IPE 240	30			
Front plate splice	IPE400 - IPE 400	31			
Front plate splice	IPE 160 - IPE 160	184			
Front plate splice	IPE 270 - IPE 270	51			
Front plate splice	IPE 720 - IPE 720	3			
Gable wall end plate	Gable wall end plate	1142			
Gusset plate at one diago	Gusset plate at one diagonal	96			
Single side end plate	Single side end plate	1			
Splice joint	HEB 400 - HEB 400	26			
Splice joint	HEM 800 - HEM 800	4			
Splice joint	IPE 180 - IPE 180	652			
Splice joint	IPE 270 - IPE 270	1			
Splice joint	IPE 400 - IPE 400	120			
Splice joint	IPE 500 - IPE 500	658			
Splice joint	IPE 720 - IPE 720	84			
Splice joint	Piastra su HEB 400	46			
Splice joint	Piastra Su HEM 800	51			
Splice joint	Piastra Su IPE 400	5			
Splice joint	Piastra Su IPE 750	5			
Splice joint	Solo Piastra Superiore Saldata	6			
Tube base plate	Tube base plate	22			
Totale generale: 4037					

## 12.3 Distinta dei bulloni

AUTODESK ADVANCE STEEL		Progetto:	Commessa:						
DISTINTA BULLONI		Cliente:	Disegnatore:	Data					
		11/04/2018							
Tipo codice	Qta	Materiale	Trattamento	Lunghezza	Tipo	Peso di pezzi (Kg/cad)	Peso Totale (kg)	Descrizione	
				(mm)					
Bullone UNI 5712 M16x55	312	10.9	Zincato	55	UNI 5712	0,2	63,3		
Bullone UNI 5712 M16x50	99	10.9	Zincato	50	UNI 5712	0,2	19,3		
Bullone UNI 5712 M16x45	846	10.9	Zincato	45	UNI 5712	0,2	158,2		
Bullone UNI 5712 M16x40	14	10.9	Zincato	40	UNI 5712	0,2	2,5		
Bullone UNI 5712 M16x320	4	10.9	Zincato	320	UNI 5712	0,1	0,3		
Bullone UNI 5712 (DIN 6914) M20x75	36	8.8	Zincato	75	UNI 5712 (DIN	0,4	13,5		
Bullone UNI 5712 (DIN 6914) M16x70	102	8.8	Zincato	70	UNI 5712 (DIN	0,2	23		
Bullone UNI 5712 (DIN 6914) M16x50	80	8.8	Zincato	50	UNI 5712 (DIN	0,2	15,6		
Bullone ASTM A325 M19.05x69.85	6	10.9	Zincato	70	ASTM A325	0,4	2,5		
Bullone ASTM A325 M19.05x63.5	277	10.9	Zincato	64	ASTM A325	0,4	111,4		
Bullone ASTM A325 M19.05x57.15	475	10.9	Zincato	57	ASTM A325	0,4	184,3		
Bullone ASTM A325 M19.05x50.8	42	10.9	Zincato	51	ASTM A325	0,4	15,7		
Bullone ASTM A325 M19.05x44.45	29	10.9	Zincato	44	ASTM A325	0,4	10,4		
							2322	619,9	
FIX Z M8X55/5-0 A4	88	A4	Zincato	55	SPIT FIX Z	0	0,2		
Cane anchor M24X500	92	10.9	Zincato	488	Cane anchor	2,4	223,6		
							180	223,7	
13918 SD S235J2G3 10 100	1	S235J2G3		100	13918 SD	0,1	0,1		
							1	0,1	
							2503	843,7	

## 12.4 Abaco dei circuiti elettrici

QE.S.P0.02 (O,P15,20), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN1	Illuminazione Sbarco Ascensori Liv +1		12	1
0.IN3	Illuminazione 1 Hall Liv +1		11	1
0.IN7	Illuminazione 1 Liv +1		10	1
0.IN9	Illuminazione 1 Liv +1		14	1
0.IN11	Illuminazione 2 Sbarco Ascensori Liv +1		13	1
0.IN12	Illuminazione 2 Hall Liv +1		11	1
P15.IR1	Illuminazione 3 Sbarco Ascensori Liv +1		9	1
P15.IR3	Illuminazione 3 Hall Liv +1		12	1
P15.IR7	Illuminazione 1 Liv +1		6	1
P15.IR8	Illuminazione 2 Liv +1		9	1
S0.IS1	Illuminazione 1 Sicurezza	720 VA	12	1
S0.IS2	Illuminazione 2 Sicurezza	1440 VA	24	1
S0.IS3	Illuminazione Sicurezza	240 VA	4	1
S0.IS4	Illuminazione Sicurezza	660 VA	11	1
QE.S.P2.01 (O,P15,20), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN1	Illuminazione Locali SX Auditorium	480 VA	9	1
0.IN2	Illuminazione Locali DX Auditorium	480 VA	9	1
0.IN3	Illuminazione 1 Locali Sud Auditorium	240 VA	6	1
0.IN6	Illuminazione 1 Auditorium LIV +3	600 VA	15	1
0.IN7	Illuminazione 2 Auditorium LIV +3	400 VA	10	1
0.IN8	Illuminazione 3 Auditorium LIV +3	320 VA	8	1
0.IN9	Illuminazione 4 Auditorium LIV +3	240 VA	6	1
0.IN10	Illuminazione 5 Auditorium LIV +3	240 VA	6	1
P15.IR1	Illuminazione Locali SX Auditorium	360 VA	6	1
P15.IR2	Illuminazione Locali DX Auditorium	360 VA	6	1
P15.IR3	Illuminazione 1 Locali Sud Auditorium	80 VA	2	1
P15.IR5	Illuminazione Locali 3 Sud Auditorium	280 VA	5	1
P15.IR6	Illuminazione Locali 4 Zona Sud Auditorium	280 VA	5	1
P15.IR7	Illuminazione 1 Auditorium LIV +3	1520 VA	25	1
P15.IR8	Illuminazione 2 Auditorium LIV +3	1520 VA	25	1
P15.IR9	Illuminazione Cattedra Auditorium	1360 VA	22	1
S0.IS3	Illuminazione Sicurezza	120 VA	3	1
QE.S.P2.02 (O,P15,20), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN1	Illuminazione 1 sala	480 VA	12	1
0.IN3	Illuminazione 2 sala	360 VA	9	1
0.IN4	Illuminazione 3 sala	360 VA	9	1
P15.IR1	Illuminazione 1 sala	480 VA	12	1

QE.S.P2.04 (O,P15,20), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN2	illuminazione 1 Corridoio	480 VA	6	1
0.IN3	illuminazione 2 Corridoio	1200 VA	17	1
0.IN4	illuminazione 3 Corridoio	480 VA	6	1
0.IN5	illuminazione 4 Corridoio	1120 VA	16	1
0.IN7	illuminazione 5 Corridoio	280 VA	7	1
0.IN10	illuminazione Sbarco Ascensori	360 VA	9	1
P15.IR2	illuminazione 1 Corridoio	480 VA	6	1
P15.IR3	illuminazione 2 Corridoio	480 VA	7	1
P15.IR4	illuminazione 3 Corridoio	440 VA	6	1
P15.IR6	illuminazione 4 Corridoio	120 VA	3	1
P15.IR8	illuminazione Sbarco Ascensori	160 VA	4	1
S0.IS1	illuminazione 1 Sicurezza	760 VA	13	1
S0.IS2	illuminazione 2 Sicurezza	760 VA	10	1
QE.S.P3.01 (O,P15,S0), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN1	illuminazione sala 1	120 VA	3	1
0.IN2	illuminazione sala 2	160 VA	4	1
0.IN3	illuminazione sala 3	120 VA	3	1
0.IN4	illuminazione sala 4	160 VA	4	1
P15.IR1	illuminazione sala 1	160 VA	4	1
P15.IR2	illuminazione sala 2	200 VA	5	1
P15.IR3	illuminazione sala 3	160 VA	4	1
P15.IR4	illuminazione sala 4	200 VA	5	1
QE.S.P3.02 (O,P15,S0), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN1	illuminazione 1 sala	480 VA	12	1
0.IN3	illuminazione 2 sala	360 VA	9	1
0.IN4	illuminazione 3 sala	320 VA	8	1
P15.IR1	illuminazione 1 sala	520 VA	13	1
QE.S.P3.03 (O,P15,S0), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN1	illuminazione 1 sala	480 VA	12	1
0.IN2	illuminazione Sala Regia	80 VA	2	1
0.IN3	illuminazione 2 sala	360 VA	9	1
0.IN4	illuminazione 3 sala	320 VA	8	1
P15.IR1	illuminazione 1 sala	520 VA	13	1
P15.IR2	illuminazione Sala Regia	80 VA	2	1
QE.S.P3.04 (O,P15,S0), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN3	illuminazione Locale Quadro	160 VA	4	1
0.IN4	illuminazione Sbarco Ascensori	200 VA	5	1
0.IN5	illuminazione Scale Sbarco	160 VA	4	1
0.IN8	illuminazione WC Uomo	980 VA	15	1
0.IN9	illuminazione WC Donna			2
0.IN10	Lampade EXIT Corridoio Uscita	480 VA	6	1
P15.IR3	illuminazione Sbarco Ascensori	280 VA	5	1
S0.IS1	illuminazione Sbarco Ascensori	400 VA	8	1

QE.S.P4.01(O,P15,S0), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN1	Illuminazione 1 Biblioteca/Mediatca	660 VA	9	1
0.IN2	Illuminazione 2 Biblioteca/Mediatca	660 VA	9	1
0.IN3	Illuminazione 1 WC e Locali Tecnici	1140 VA	19	1
0.IN4	Illuminazione 2 WC e Locali Tecnici	920 VA	15	1
P15.IR1	Illuminazione 1 Biblioteca/Mediatca	300 VA	4	1
P15.IR2	Illuminazione 2 Biblioteca/Mediatca	400 VA	5	1
P15.IR3	Illuminazione 1 WC e Locali Tecnici	500 VA	8	1
S0.IS1	Illuminazione Sicurezza			2
S0.IS2	Scorta	80 VA	1	1

QE.S.P4.02(O,P15,S0), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN1	Illuminazione 1 Caffetteria	240 VA	3	1
0.IN2	Illuminazione 2 Caffetteria	240 VA	3	1
0.IN5	Illuminazione Cucina e Deposito	480 VA	8	1
P15.IR1	Illuminazione 1 Caffetteria	240 VA	3	1
P15.IR2	Illuminazione 2 Caffetteria	160 VA	2	1
P15.IR3	Illuminazione Cucina e Deposito	240 VA	4	1
S0.IS1	Illuminazione Sicurezza	560 VA	9	1
S0.IS2	Scorta	160 VA	2	1

QE.S.P4.03.(O,P15,S0), 230 V/400 V, Tre Fase, 4 Cavi, Raccordo a Y				
0.IN3	Illuminazione Locale Tecnico	1920 VA	24	1
0.IN4	Illuminazione Corridoio SX	400 VA	5	1
0.IN5	Illuminazione Corridoio DX	520 VA	7	1
0.IN9	Illuminazione Connettivo	200 VA	5	1
0.IN10	Illuminazione 1 Corridoio	320 VA	8	1
0.IN11	Illuminazione 2 Corridoio	640 VA	8	1
0.IN12	Illuminazione 3 Corridoio	320 VA	8	1
0.IN13	Illuminazione 4 Corridoio	640 VA	8	1
P15.IR3	Illuminazione Locale Tecnico	640 VA	8	1
P15.IR4	Illuminazione Corridoio SX	320 VA	4	1
P15.IR5	Illuminazione Corridoio DX	320 VA	4	1
P15.IR6	Illuminazione Connettivo	120 VA	3	1
P15.IR9	Illuminazione 1 Corridoio	880 VA	15	1
P15.IR10	Illuminazione 2 Corridoio	880 VA	15	1
S0.IS2	Illuminazione Corridoio	160 VA	2	1
S0.IS3	Illuminazione Corridoio	160 VA	2	1

Totale generale: 125

## 12.5 Consumi Applicazione

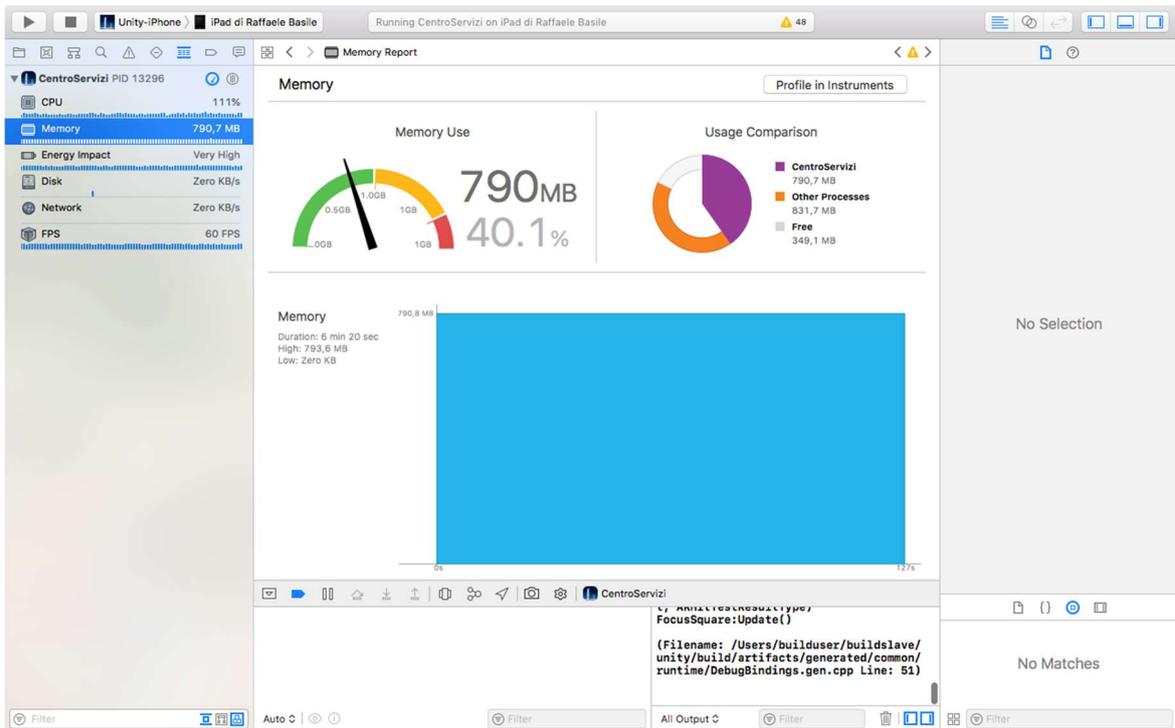


Figura 106: Utilizzo RAM

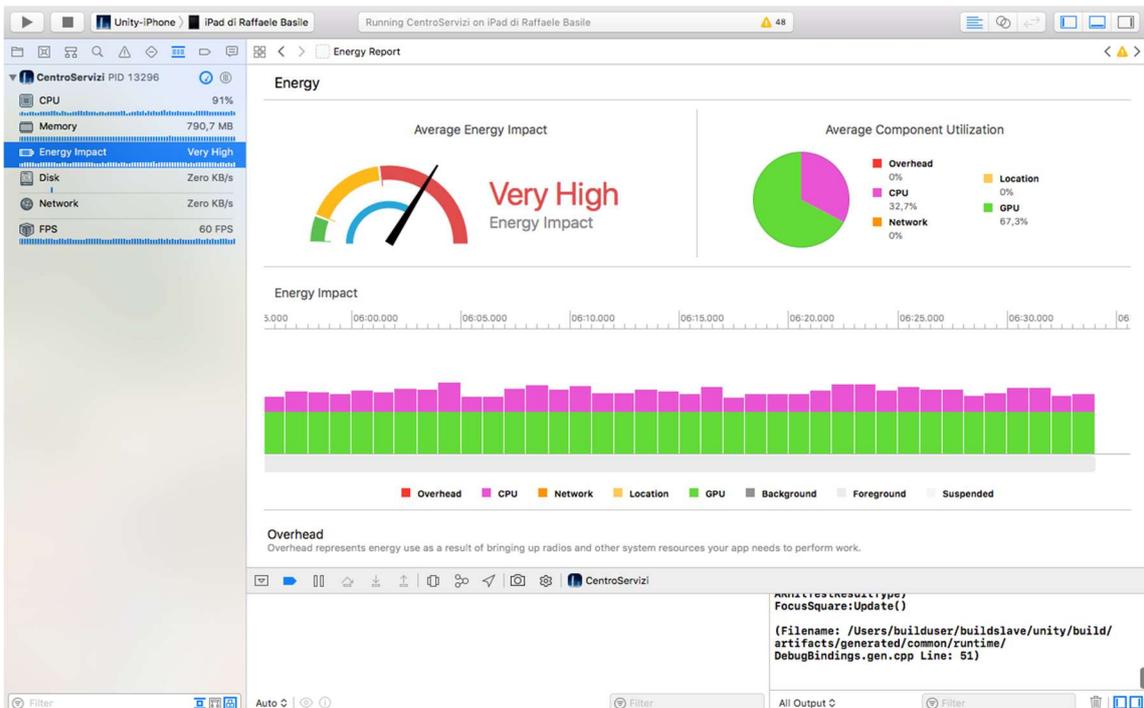


Figura 107: Utilizzo Batteria

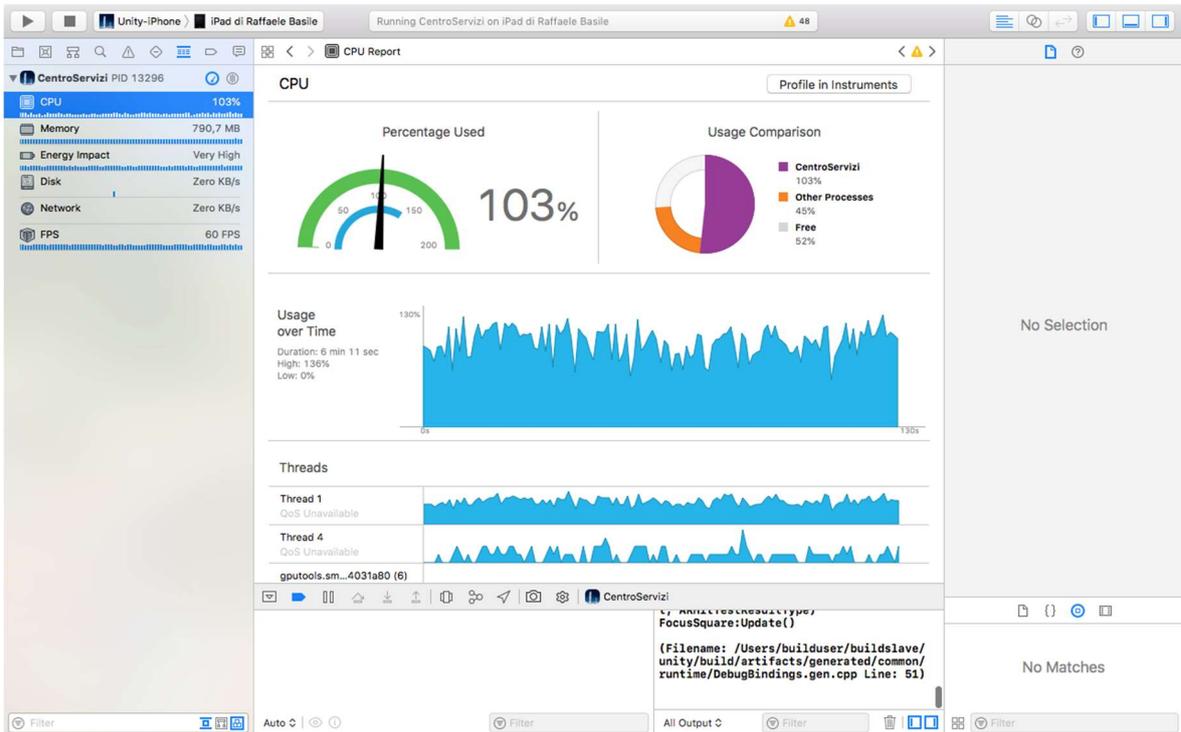


Figura 109: Utilizzo CPU

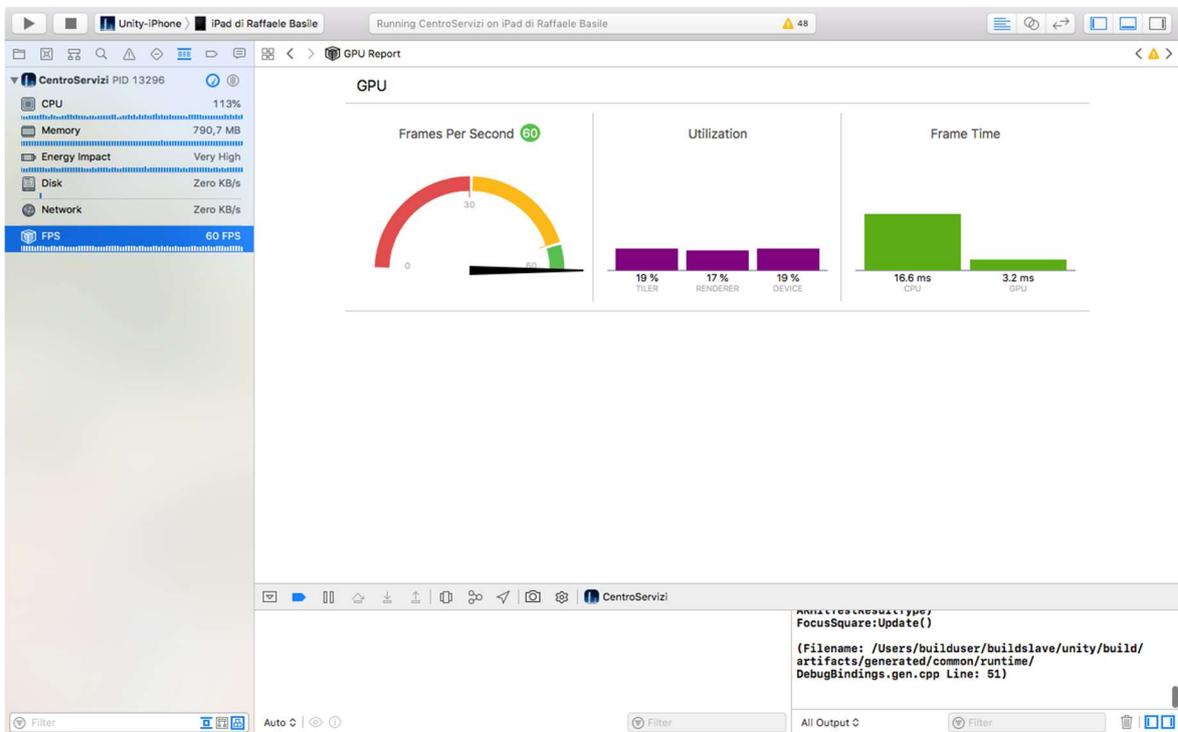


Figura 108: Frames per Second

## 12.6 UnityARHitTestExample.cs

```
1 using System;
2 using System.Collections.Generic;
3 using UnityEngine.Events;
4
5 namespace UnityEngine.XR.iOS
6 {
7     public class UnityARHitTestExample : MonoBehaviour
8     {
9         public Transform m_HitTransform;
10        public float maxRayDistance = 30.0f;
11        public LayerMask collisionLayer = 1 << 10; //ARKitPlane layer
12
13
14
15        bool HitTestWithResultType (ARPoint point, ARHitTestResultType resultTypes)
16        {
17            List<ARHitTestResult> hitResults = UnityARSessionNativeInterface.GetARSessionNativeInterface ().HitTest (point, resultTypes);
18            if (hitResults.Count > 0) {
19                foreach (var hitResult in hitResults) {
20                    Debug.Log ("Got hit!");
21                    m_HitTransform.position = UnityARMatrixOps.GetPosition (hitResult.worldTransform);
22                    m_HitTransform.rotation = UnityARMatrixOps.GetRotation (hitResult.worldTransform);
23                    Debug.Log (string.Format ("x:{0:0.#####} y:{1:0.#####} z:{2:0.#####}",
24                        m_HitTransform.position.x, m_HitTransform.position.y, m_HitTransform.position.z));
25                    return true;
26                }
27            }
28            return false;
29        }
30
31
32
33        private bool IsPointerOverUIObject()
34        {
35            PointerEventData eventDataCurrentPosition = new PointerEventData(EventSystem.current);
36            eventDataCurrentPosition.position = new Vector2(Input.mousePosition.x, Input.mousePosition.y);
37            List<RaycastResult> results = new List<RaycastResult>();
38            return results.Count > 0;
39        }
40
41        // Update is called once per frame
42        void Update () {
43
44            #if UNITY_EDITOR //we will only use this script on the editor side, though there is nothing that would prevent it from working on device
45            if (Input.GetMouseButtonDown (0)) {
46                Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);
47                RaycastHit hit;
48
49                //we'll try to hit one of the plane collider gameobjects that were generated by the plugin
50                //effectively similar to calling HitTest with ARHitTestResultType.ARHitTestResultTypeExistingPlaneUsingExtent
51                if (Physics.Raycast (ray, out hit, maxRayDistance, collisionLayer)) {
52                    //we're going to get the position from the contact point
53                    m_HitTransform.position = hit.point;
54                    Debug.Log (string.Format ("x:{0:0.#####} y:{1:0.#####} z:{2:0.#####}", m_HitTransform.position.x, m_HitTransform.position.y, m_HitTransform.position.z));
55
56                    //and the rotation from the transform of the plane collider
57                    m_HitTransform.rotation = hit.transform.rotation;
58                }
59            }
60            #else
61            if (Input.touchCount > 0 && m_HitTransform != null)
62            {
63                var touch = Input.GetTouch(0);
64                if ((touch.phase == TouchPhase.Began || touch.phase == TouchPhase.Moved) && !IsPointerOverUIObject() && !EventSystem.current.IsPointerOverGameObject(0))
65                {
66                    var screenPosition = Camera.main.ScreenToViewportPoint(touch.position);
67                    ARPoint point = new ARPoint {
68                        x = screenPosition.x,
69                        y = screenPosition.y
70                    };
71
72                    // prioritize results types
73                    ARHitTestResultType[] resultTypes = {
74                        ARHitTestResultType.ARHitTestResultTypeExistingPlaneUsingExtent,
75                        // if you want to use infinite planes use this:
76                        //ARHitTestResultType.ARHitTestResultTypeExistingPlane,
77                        ARHitTestResultType.ARHitTestResultTypeHorizontalPlane,
78                        ARHitTestResultType.ARHitTestResultTypeFeaturePoint
79                    };
80
81                    foreach (ARHitTestResultType resultType in resultTypes)
82                    {
83                        if (HitTestWithResultType (point, resultType))
84                        {
85                            return;
86                        }
87                    }
88                }
89            }
90            #endif
91        }
92
93    }
94 }
95 }
96
97
```

Figura 110: UnityARHitTestExample.cs

Sono state evidenziate in rosso le modifiche apportate allo script originale. Queste modifiche impediscono di resettare l'origine del modello quando si tocca un pulsante posizionato sullo schermo.

## 12.7 Clock.cs

```
1 using UnityEngine;
2 using UnityEngine.UI;
3 using System.Collections;
4 using System;
5 public class Clock : MonoBehaviour
6 {
7     #region Declarations
8     public static float unscaledTime; //refer to this value when determining timed-event checks (measures playtime in seconds since the start of the g
9     public static float time; //refer to this value "Time.timeElapsed" when determining in-game timed-event checks such as weather (measures time in s
10    [SerializeField]
11    private Text clockText;
12
13    [SerializeField]
14    private Text dateText;
15    [SerializeField]
16    private float clockSpeed = 1f;
17    DateTime currentTime;
18    DateTime deadlineTime;
19    public int howMuchMonth;
20    private int difference;
21
22    #endregion
23
24    public Text dateMaintenance;
25    public Text maintenanceDeadline;
26    public int dateConverter;
27    private int dateConverterDAY;
28    private int dateConverterMONTH;
29    private int dateConverterYEAR;
30    private int currentdate;
31    private int currentdateDAY;
32    private int currentdateMONTH;
33    private int currentdateYEAR;
34
35    void Start() {
36
37        currentTime = System.DateTime.Now;
38
39        #region date converter
40        currentdateDAY = int.Parse(currentTime.ToString("dd"));
41        currentdateMONTH = int.Parse(currentTime.ToString("MM"));
42        currentdateYEAR = int.Parse(currentTime.ToString("yyyy"));
43        currentdate = currentdateDAY + (currentdateMONTH * 30) + (currentdateYEAR * 365);
44        Debug.Log(currentdate);
45    #endregion
46
47    dateMaintenance.text = PlayerPrefs.GetString(GetInstanceId() + " " + "dateMaintenance", currentTime.ToString("dd/MM/yyyy").ToString());
48    maintenanceDeadline.text = PlayerPrefs.GetString(GetInstanceId()+" " + "maintenanceDeadline", deadlineTime.ToString("dd/MM/yyyy").ToString());
49
50    #region Clock
51    void Awake()
52    {
53        Time.timeScale = 1.0f;
54        InvokeRepeating("Lapse", 0, clockSpeed * Time.timeScale);
55    }
56    void Lapse()
57    {
58        currentTime = System.DateTime.Now;
59
60        clockText.text = currentTime.ToString("hh:mm:ss");
61        dateText.text = currentTime.ToString("dd/MM/yyyy"); //Specifies the name of the day, its date, the name of the month, and the year);
62    }
63
64    public void NormalSpeed()
65    {
66        Time.timeScale = 1.0f;
67    }
68    void MinuteSpeed()
69    {
70        Time.timeScale = 60.0f;
71    }
72    void HourSpeed()
73    {
74        Time.timeScale = 600.0f;
75    }
76
77    public void maintenanceCommand() {
78
79        dateMaintenance.text = currentTime.ToString("dd/MM/yyyy");
80        deadlineTime = currentTime.AddMonths(howMuchMonth);
81
82        #region date converter
83        dateConverterDAY = int.Parse(deadlineTime.ToString("dd"));
84        dateConverterMONTH = int.Parse(deadlineTime.ToString("MM"));
85        dateConverterYEAR = int.Parse(deadlineTime.ToString("yyyy"));
86        dateConverter = dateConverterDAY + (dateConverterMONTH * 30) + (dateConverterYEAR * 365);
87        Debug.Log(dateConverter);
88    #endregion
89
90    maintenanceDeadline.text = deadlineTime.ToString("dd/MM/yyyy");
91
92    PlayerPrefs.SetString(GetInstanceId() + " " + "dateMaintenance", currentTime.ToString("dd/MM/yyyy"));
93    PlayerPrefs.SetString(GetInstanceId()+" " + "maintenanceDeadline", deadlineTime.ToString("dd/MM/yyyy"));
94    }
95
96    #endregion
97 }
98
99 }
```

Figura 111: Script Clock.cs

## 12.8 MenuManager.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using System;
6 using System.Xml.Linq;
7 using SimpleJSON;
8 using System.Globalization;
9 using UnityEngine.Networking;
10 using UnityEngine.EventSystems;
11
12 public class MenuManager : MonoBehaviour {
13
14     public Text dateMaintenance;
15     public Text maintenanceDeadline;
16     private DateTime currentTime;
17     private DateTime deadlineTime;
18     public string MenuName;
19     public string ObjName;
20     public int howMuchTime;
21
22     private static readonly string baseUrl = "http://tesingegneriaedilepolito.altervista.org";
23     private static readonly string auth = "8a47woy5adsqsed45sdo8y378oa8";
24
25
26     // Use this for initialization
27     void Start () {
28         currentTime = System.DateTime.Now;
29
30         dateMaintenance.text = "Caricamento...";
31         maintenanceDeadline.text = "Caricamento...";
32
33         ObjName = this.transform.parent.parent.name;
34         Debug.Log("ObjName = " + ObjName);
35         MenuName = this.name;
36         Debug.Log("MenuName = " + MenuName);
37
38         Load(ObjName, MenuName, (dateA, dateB) => {
39             Debug.Log("loading " + ObjName + " " + MenuName);
40             // Clock.maintenanceDeadline.text = dateA.ToString("dd/MM/yyyy");
41             // dateA e dateB contengono le due date per oggetto1 menu1
42             // questa porzione di codice viene eseguita se è tutto ok
43             dateMaintenance.text = dateA.ToString("dd/MM/yyyy");
44             maintenanceDeadline.text = dateB.ToString("dd/MM/yyyy");
45         }, () => {
46             //questa porzione di codice invece viene eseguita al posto di quella sopra
47             //nel caso in cui il recupero non abbia avuto successo
48             Debug.Log("error-not found");
49             dateMaintenance.text = maintenanceDeadline.text = "Manutenzione mai avvenuta";
50
51         });
52     }
53
54 }
55
56 // Update is called once per frame
57 void Update () {
58
59 }
60
61 public void maintenanceCommand()
62 {
63     currentTime = DateTime.Now;
64     deadlineTime = currentTime.AddMonths(howMuchTime);
65     dateMaintenance.text = maintenanceDeadline.text = "Impostazione...";
66     Save(ObjName, MenuName, currentTime, deadlineTime, (success) =>
67     {
68         if (success)
69         {
70             dateMaintenance.text = currentTime.ToString("dd/MM/yyyy");
71             maintenanceDeadline.text = deadlineTime.ToString("dd/MM/yyyy");
72         }
73         else
74         {
75             dateMaintenance.text = maintenanceDeadline.text = "Errore!";
76         }
77     });
78 }
79
80 }
```

```

81 public void Load(string objID, string menuID,
82                 Action<DateTime, DateTime> callback, Action errorCallback)
83 {
84     Debug.Log("Begin load phase --- data: " + objID + ", " + menuID);
85
86     StartCoroutine(RequestHTTP("load.php", "objID=" + objID + "&menuID=" + menuID, data => {
87         var response = JSON.Parse(data);
88         if (!response["success"].AsBool)
89         {
90             errorCallback(); return;
91         }
92
93         string format = "yyyy-MM-dd";
94         try
95         {
96
97             DateTime dateA = DateTime.ParseExact(response["data"]["dateA"].Value,
98                                                   format, CultureInfo.InvariantCulture);
99
100            DateTime dateB = DateTime.ParseExact(response["data"]["dateB"].Value,
101                                                  format, CultureInfo.InvariantCulture);
102
103
104            callback(dateA, dateB);
105        }
106        catch (Exception e)
107        {
108            Console.WriteLine(e);
109            errorCallback();
110            return;
111        }
112    }));
113 }
114
115 public void Save(string objID, string menuID,
116                 DateTime dateA, DateTime dateB,
117                 Action<bool> success)
118 {
119     Debug.Log("Begin save phase");
120     StartCoroutine(RequestHTTP("save.php", "objID=" + objID + "&menuID=" + menuID +
121                               "&dateA=" + dateA.ToString("yyyy-MM-dd") +
122                               "&dateB=" + dateB.ToString("yyyy-MM-dd"), (s) =>
123     {
124
125         var response = JSON.Parse(s);
126
127         Console.WriteLine("saving: " + s);
128         success(response["success"].AsBool);
129     }));
130 }
131
132 private IEnumerator RequestHTTP(string url, string data, Action<string> callback)
133 {
134     using (UnityWebRequest www = UnityWebRequest.Get(baseUrl + "/" + url + "?auth=" + auth + "&" + data))
135     {
136         yield return www.SendWebRequest();
137
138         if (www.isNetworkError || www.isHttpError)
139         {
140             callback("{success: false, error: \"script_failure\"}");
141             Debug.Log(www.error);
142         }
143         else
144         {
145             string b = www.downloadHandler.text;
146             callback(b);
147         }
148     }
149 }
150 }
151 }
152

```

Figura 112: Script Menumanager.cs

## 12.9 SCRIPT PHP

### 12.9.1 db.php

```
<?php

$host = '127.0.0.1';
$db = 'my_tesingegneriaedilepolito';
$user = 'tesingegneriaedilepolito';

register_shutdown_function( "fatal_handler" );
ob_start();

header('Content-Type: application/json');

if (!isset($_GET['auth']) || $_GET['auth'] != '8a47woy5adsqsed45sdo8y378oa8') {
    http_response_code (403);
    err("unauthorized");
}

$dns = "mysql:host=$host;dbname=$db;charset=utf8mb4";
$opt = [
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false,
];
$db = new PDO($dns, $user, "", $opt);

function err($val) {
    ob_end_clean();
    header('Content-Type: application/json');
    die(json_encode(['success' => false, 'error' => $val]));
}

function getVal($name) {
    return isset($_GET[$name]) && $_GET[$name] != " ?
        $_GET[$name] : err("missing value: " . $name);
}

function runSQL($sql, $data, $cb) {
    global $db;
    $stmt = $db->prepare($sql);
    $stmt->execute($data);
    $cb($stmt->fetch());
}

function fatal_handler() {
```

```

        $e = error_get_last();
        if($e !== NULL) {
            err(ob_get_contents() . $e["message"]);
        }
    }

function validateDate($date)
{
    $format = 'Y-m-d';
    $d = DateTime::createFromFormat($format, $date);
    return $d ? $d->format($format) : err("invalid_date");
}

```

## 12.9.2 load.php

```

<?php

require('db.php');

$data = [
    "objID" => getVal('objID'),
    "menuID" => getVal('menuID')
];

$sql = "SELECT * FROM data WHERE objID = :objID AND menuID = :menuID";

runSQL($sql, $data, function ($data) {
    if ($data) {
        die(json_encode(["success" => true, "data" => $data]));
    } else {
        err("not_found");
    }
});

```

## 12.9.3 save.php

```

<?php

require('db.php');

$dateA = validateDate(getVal('dateA'));
$dateB = validateDate(getVal('dateB'));

$data = [
    "objID" => getVal('objID'),
    "menuID" => getVal('menuID'),
    "dateA" => $dateA,
    "dateAA" => $dateA,
    "dateB" => $dateB,

```

```
"dateBB" => $dateB
];

$sql = "INSERT INTO data (objID, menuID, dateA, dateB)
VALUES (:objID, :menuID, :dateA, :dateB)
ON DUPLICATE KEY UPDATE dateA=:dateAA, dateB=:dateBB";

runSQL($sql, $data, function () {
    die(json_encode(["success" => true]));
});
```