

JPAD setup

Breve guida introduttiva alla libreria java JPAD e all'organizzazione e gestione del lavoro condiviso.

*Java toolchain of
Programs for
Aircraft Design*

SOMMARIO

1. INTRODUZIONE A JPAD	2
2. IL LINGUAGGIO DI PROGRAMMAZIONE	5
3. LAVORO IN CONDIVISIONE	7
4. SETUP DI JPAD	10
5. GESTIONE DELLE GUI	15
6. RIEPILOGO STEP PER INIZIO LAVORO SU JPAD	15

1. Introduzione a JPAD

La libreria java **JPAD** (*Java toolchain of Programs for Aircraft Design*), si configura come un ambiente software integrato in grado di effettuare uno studio su un velivolo completo o sui suoi componenti, utile in fase di analisi preliminare nell'ambito dell'*Aircraft Design*. JPAD consente di effettuare differenti tipi di analisi, anche con diversi metodi, raggruppabili nelle seguenti cinque macro aree:

- **Pesi:** Modulo che consente la stima del *breakdown* dei pesi a partire da un MTOW di primo tentativo e dai *requirements* di missione.
- **Centraggio:** Modulo per la stima della posizione del centro di gravità del velivolo e dei componenti.
- **Aerodinamica e Stabilità:** Permette la valutazione di tutte le caratteristiche aerodinamiche dei componenti e del velivolo stesso nelle varie condizioni di volo richieste. Il modulo consente il calcolo della stabilità e permette la valutazione dei relativi parametri anche considerando gli effetti non lineari.
- **Prestazioni:** Valuta i parametri di performance come il diagramma *Payload-Range*, il profilo di missione, l'inviluppo di volo, le prestazioni di decollo, di atterraggio e di salita e il diagramma di crociera.
- **Costi:** Modulo che valuta i costi diretti di esercizio (DOC) e il relativo *breakdown*.

A partire dalla definizione di un velivolo di input, delle condizioni operative, e di alcuni dati di analisi, tramite l'esecuzione di JPAD, è possibile ottenere in pochi minuti un'analisi completa di avvanprogetto.

La struttura dei file di input è stata pensata in modo da permettere all'utente di avere un semplice controllo sui dati e sulle analisi che si vogliono effettuare. Il risultato di questo studio preliminare è una struttura di file XML che rispetta la struttura modulare di un velivolo.

L'XML è un metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio marcatore basato su un meccanismo sintattico che consente di definire i nomi delle etichette, chiamate *tag*, senza dover ricorrere a parole chiave predefinite, rendendo così di semplice lettura il testo. I *tag* possono contenere informazioni in due modi: attraverso dei parametri oppure racchiudendo in sé dati o altri *tag*. Benché sia possibile assegnare un qualsivoglia nome arbitrario ai *tag*, affinché essi siano ben formati, occorre rispettare le seguenti regole:

- Ciascun documento XML deve contenere un unico elemento di massimo livello (*root*) che contenga tutti gli altri elementi del documento; cioè un tag principale che contiene tutti gli altri.

- Le sole parti di XML che possono trovarsi all'esterno del suddetto elemento sono i commenti e le direttive di elaborazione (per esempio, la dichiarazione della versione di XML).
- Ogni elemento deve avere un *tag* di apertura e uno di chiusura. I nomi dei tag sono senza spazi, uguali per apertura e chiusura, fatta eccezione per il / che indica un tag di chiusura.

<nome_tag eventuale_attributo="val"> valore </nome_tag>

- Se un tag è vuoto (i.e. senza alcun valore) può prevedere la forma abbreviata (terminando il *tag* di apertura con la stringa di caratteri "</>".)
- Un *tag* può iniziare con una lettera o un *underscore*, e può contenere lettere, numeri, punti, l'*underscore* o trattini. Non sono ammessi spazi o altri caratteri.
- Gli elementi devono essere opportunamente nidificati, cioè i *tag* di chiusura devono seguire l'ordine inverso dei rispettivi *tag* di apertura.
- XML fa distinzione tra maiuscole e minuscole, per cui i nomi dei *tag* e degli attributi devono coincidere nei *tag* di apertura e chiusura anche in relazione a questo aspetto.
- I valori degli attributi devono sempre essere racchiusi tra singoli o doppi apici.
- Infine, ogni documento deve contenere un prologo, ossia una prima riga, che lo identifichi come XML, ne specifichi la versione, e riporti una codifica per la corretta interpretazione dei caratteri. Questo sarà fuori dal tag principale di massimo livello.

Ad esempio:

<?xml version="1.0" encoding="utf-8"?>

- In un documento XML è possibile inserire commenti racchiusi dalle sequenze "<!--" e "-->", i quali saranno ignorati dal software durante la lettura del file.

Per quanto riguarda l'input di JPAD, è possibile individuare due macro blocchi di documenti XML: uno riferito alla definizione del velivolo, e un altro dedito ai dati necessari alle analisi. I file sono organizzati secondo la struttura ad albero mostrata nella parte a sinistra in Figura 1. Nella medesima figura è rappresentato lo schema delle analisi di JPAD e gli eventuali output.

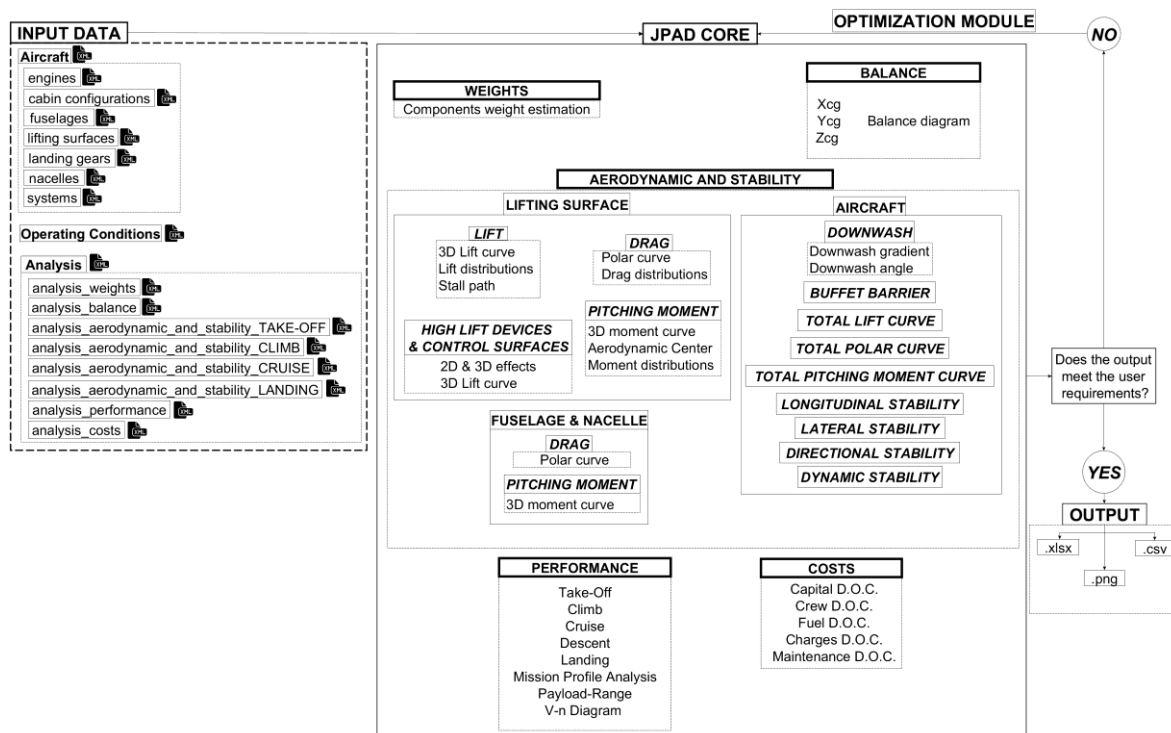


Figura 1 : Flow chart delle analisi in JPAD

JPAD è dotato di un'interfaccia grafica in fase di sviluppo per l'immissione dei dati e la gestione delle analisi, mostrata in Figura 2.

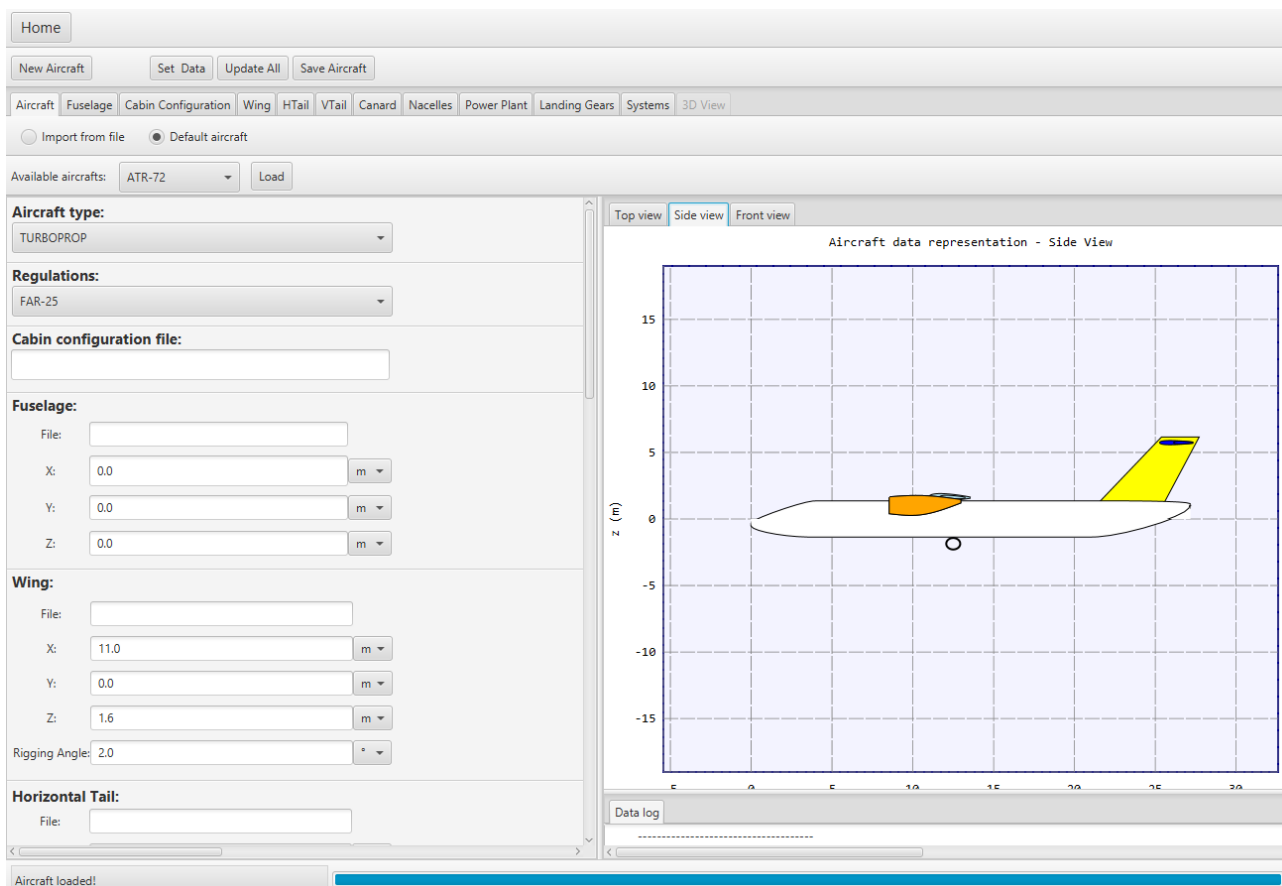


Figura 2: Interfaccia grafica (GUI) di JPAD per l'immissione dei dati di input.

2. Il linguaggio di programmazione

Java è un linguaggio di programmazione ad alto livello, orientato agli oggetti e a tipizzazione statica, specificatamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione.

Prima di iniziare è necessario avere una conoscenza, seppur base, del linguaggio di programmazione JAVA. Si può far riferimento alle seguenti fonti:

- playlist youtube → <https://www.youtube.com/playlist?list=PL0qAPtx8YtJe2dpE7di4aPJwrQuRD6lDD>
- guida → <https://drive.google.com/open?id=1SvSrQYTVQV34V5GI8ziTQ5Ava5rRzk2oW>

Installazione di JAVA

Se non si dispone ancora di java installato sul pc, occorre scaricare il jdk e il jre della versione che attualmente si sta utilizzando (al 05/2018 - JAVA 8, v.171).

È possibile scaricare java al seguente link, in automatico saranno installati sia jdk che jre:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

accettare il contratto di licenza e scaricare la versione adatta al proprio sistema operativo.

Una volta scaricato java occorre installarlo seguendo il wizard. A questo punto, bisogna immettere il jdk e il jre nelle variabili d'ambiente relative al sistema, all'interno della variabile *Path*.

Variabili d'ambiente relative al sistema → path → nuova variabile

- C:\Program Files\Java\jdk1.8.0_151\bin
- C:\Program Files\Java\jre1.8.0_151\bin

Aggiornamento di JAVA

importante che tutti gli sviluppatori siano aggiornati sull'ultima versione di java altrimenti si incorre in problemi durante il lavoro condiviso. Le versioni pari di java non sono da noi utilizzate (spesso presentano problemi), pertanto l'aggiornamento sarà fatto solo su quelle dispari.

Dopo la prima installazione di java, essendoci un'unica versione installata sul pc, essa sarà l'unica a cui il SO farà riferimento. Quando si installa una nuova versione di java sul pc per aggiornamento, occorre assicurarsi che quest'ultima sia quella effettivamente utilizzata.

Per fare ciò, dopo l'installazione dell'ultima versione, occorre cambiare la cartella cui punta la variabile d'ambiente *Path* relativamente al jdk e al jre, modificando, ad esempio per passare dalla versione 151 alla 153:

C:\Program Files\Java\jdk1.8.0_151\bin → C:\Program Files\Java\jdk1.8.0_153\bin

A questo punto per accertarsi di quale sia la versione di utilizzo, andare in Configura java (dalla ricerca windows scrivere "java"), **Errore. L'origine riferimento non è stata trovata..**

Configura java → Java → visualizza → accertarsi che ci sia l'ultima versione e che sia abilitata

In qualsiasi momento, per vedere qual è la versione di corrente utilizzo, dal prompt dei comandi, digitare il comando `java -version`, Figura 4.

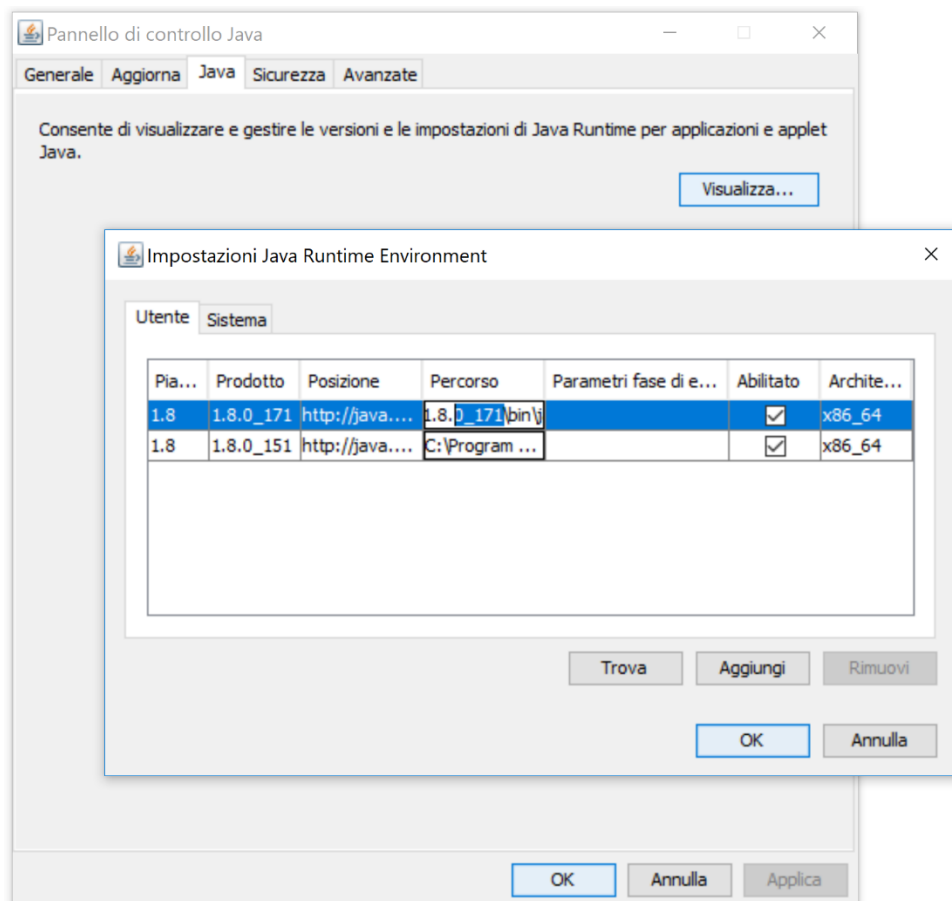


Figura 3: Settaggio versione di java utilizzata

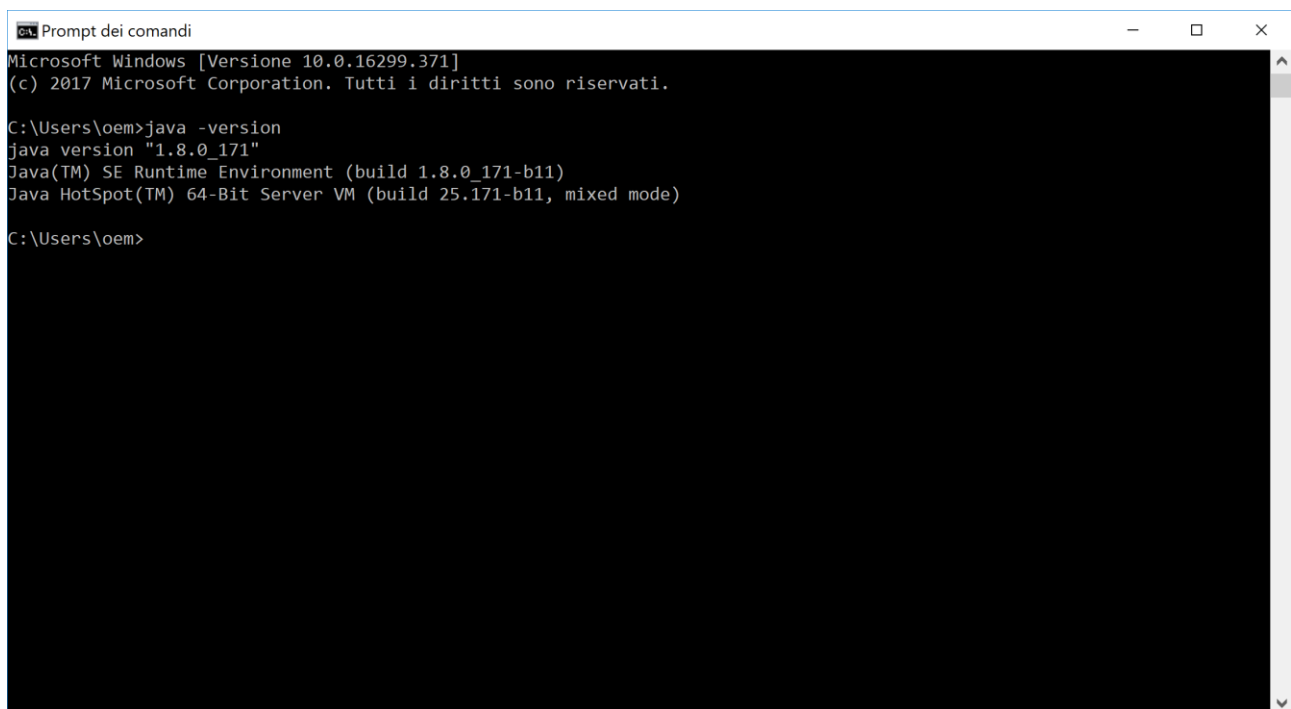


Figura 4: Verifica versione di java di corrente utilizzo.

3. Lavoro in condivisione

Al fine di poter lavorare più persone contemporaneamente sullo stesso progetto, occorre che lo stesso sia disponibile su una piattaforma online, e che sia sempre aggiornato con l'ultima versione. È chiaro che, al fine di non produrre versioni in conflitto tra di loro, non bisognerebbe lavorare sulla stessa classe (i.e. sullo stesso file) in contemporanea. Al termine del lavoro, occorrerà fare l'update delle modifiche, cosicché la versione online sarà sempre aggiornata. Prima di iniziare il lavoro, ogni volta, sarà necessario fare quindi il download dell'ultima versione più aggiornata.

Registrazione a Github

La piattaforma sulla quale attualmente è in condivisione la repository online è github. Per prima cosa occorre registrarsi al sito <https://github.com/> creando un proprio account.

Bash di GIT per la gestione della repository

Dopo aver ottenuto un account è utile scaricare la Git bash, ossia una finestra di comando per gestire i file posti in condivisione sulla piattaforma git hub, al seguente link: <https://git-scm.com/downloads> . Si consiglia di lasciare tutte le opzioni di default tranne quella in Figura 5.

La bash di Git non è l'unico modo per accedere alla repository online. È possibile farlo anche tramite altre finestre di comando come **ConEmu**, il quale non sarà approfondito in questa sede.

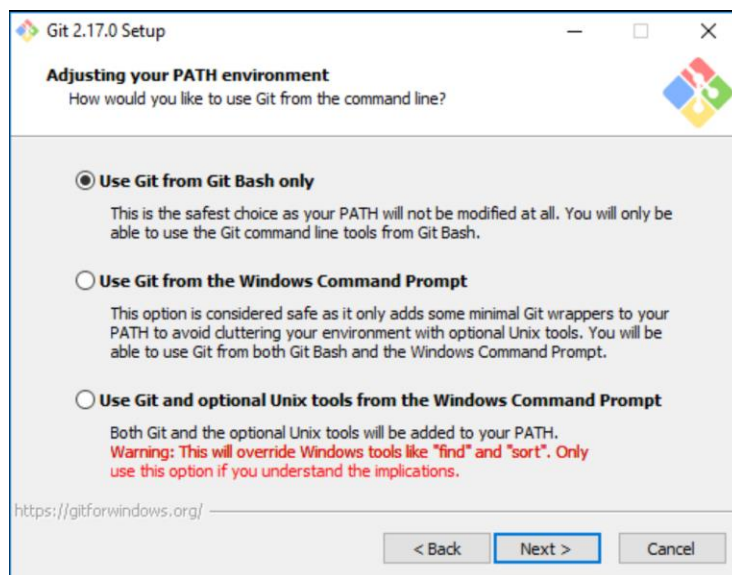


Figura 5: Opzione consigliata nell'istallazione delle bash di Git.

Download della repository

Il link della repository online di jpad è <https://github.com/Aircraft-Design-UniNa/jpad>

La prima cosa da fare è un clone della repository online, ossia scaricare una copia locale della repository sul proprio pc sulla quale si lavorerà di volta in volta. Questa copia locale sarà aggiornata scaricando dalla repository online soltanto le modifiche che gli altri utenti hanno effettuato (tramite un comando chiamato **pull**), mentre la versione Master online sarà a sua volta modificata dagli utenti che lavorano sulle varie parti mediante i **commit** degli stessi (ossia un upload delle sole modifiche tramite il comando **push**). Chiaramente dopo che sono state effettuate delle modifiche, e queste sono salvate tramite il **commit**, semplicemente con un **push** verrà in automatico fatto un upload delle sole modifiche, così come il **pull** farà in automatico il download delle sole modifiche. Si ribadisce, quindi, che è una best practice fare un **pull** prima di lavorare al

progetto e un **push** subito dopo aver finito cosicché sia la versione Master online sia sempre aggiornata e ciò permette agli altri sviluppatori di lavorarci, sia la versione locale sul proprio PC.

Il primo download della repository sarà, ovviamente, il più lungo, in quanto è necessario scaricarla completamente. È bene fare il clone della repository (i.e. download) tramite protocollo SSH, essendo più sicuro del HTTPS.

Protocollo SSH

Per fare ciò occorre generare una chiave SSH se non se ne possiede già una.

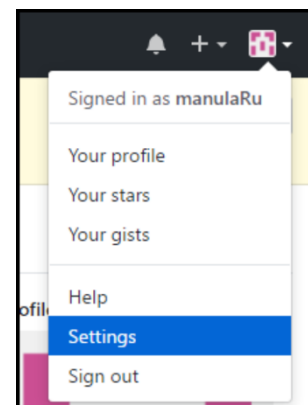
- Aprire la Git Bash
- Scrivere il comando per la generazione della chiave, inserendo la propria mail di GitHub
ssh-keygen -t rsa -b 4096 -C "*inserire la propria mail*"
- Cliccare invio
- Inserire una password e confermarla

A questo punto sarà generata la chiave SSH, come si vede in Figura 6 e bisogna aggiungerla.

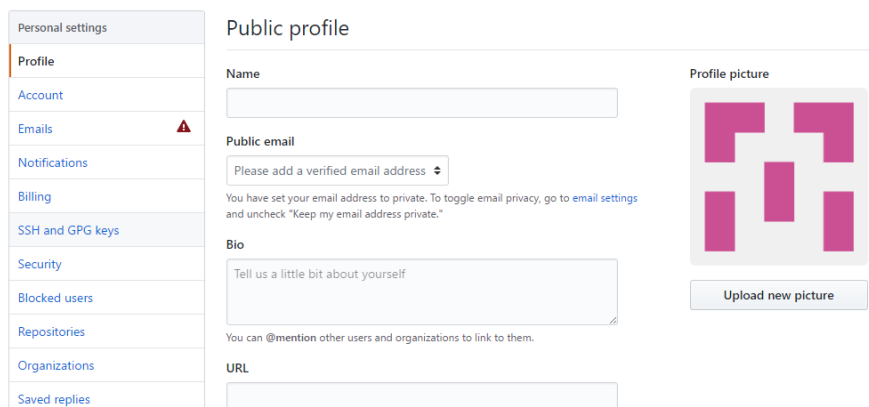
- Nella Git Bash scrivere il comando **eval \$(ssh-agent -s)**
- Infine **ssh-add ~/.ssh/id_rsa** e mettere la password scelta

Bisogna ora aggiungerla al proprio account

- Per copiare la chiave ssh basta eseguire il seguente comando nella bash **clip < ~/.ssh/id_rsa.pub**
- Infine bisogna aggiungere la chiave nel proprio account.
 - Entrare nel proprio account Github
 - Cliccare in alto a destra sulla foto profilo e andare in Settings



- Cliccare su SSH and GPG keys e su New SSH key



- In Titolo immettere un qualsivoglia nome, mentre in "Key" incollare la chiave ssh che è stata copiata tramite la git bash con il precedente comando.

- Cliccare su “Add SSH key”

```

MINGW64:/c/Users/Marola
Marola@DESKTOP-GIR1Q8F MINGW64 ~
$ ssh-keygen -t rsa -b 4096 -C "manuela.ruocco@alice.it"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Marola/.ssh/id_rsa):
Created directory '/c/Users/Marola/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Marola/.ssh/id_rsa.
Your public key has been saved in /c/Users/Marola/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:5pfVF0SiTXF1UZovYHaqkok2H9Rcf6LxTx8VpNKIH0g manuela.ruocco@alice.it
The key's randomart image is:
+---[RSA 4096]-----+
|  .. +o%|
|  .. o+o+=. |
|  . o.B.B. |
|  Eo + B o. |
|  S o + + + |
|  = o + + + |
|  + + + . . |
|  . o +   o |
|  .   .   . |
+---[SHA256]-----+
Marola@DESKTOP-GIR1Q8F MINGW64 ~
$ |

```

Figura 6: Generazione della chiave per protocollo SSH.

Clone della directory

A questo punto è possibile fare il primo clone della directory.

- Dalla Git bash, andare nel percorso dove si vuole salvare la cartella da clonare. Ad esempio, per salvarlo in C:\Users\nomeUtente
cd /c/Users/nomeUtente
- A questo punto occorre salvare il link dalla directory di jpad per fare il clone. Alla pagina <https://github.com/Aircraft-Design-UniNa/jpad> cliccare su *clone or download* e poi use SSH. Copiare il link come in Figura 7.
- Dalla Git bash, nella directory selezionata, tramite il comando clone, fare il download della repository. Viste le grandi dimensioni di jpad, si consiglia di avere una buona connessione e tempo a sufficienza.
git clone _incollareLinkCopiato

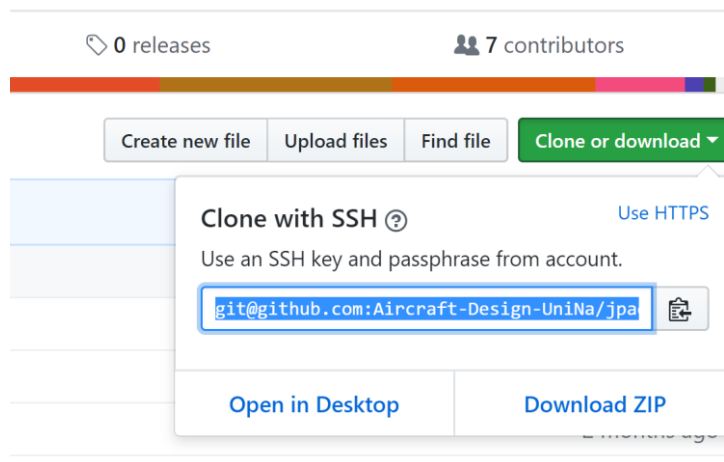


Figura 7: Link per il clone tramite protocollo SSH.

LFS per file di grandi dimensioni

Per la gestione di file di grandi dimensioni è possibile utilizzare il protocollo LFS (Large File Storage) disponibile al seguente indirizzo. <https://git-lfs.github.com/> . Dopo aver creato una repository è possibile, per rendere più veloci i successivi download e upload, utilizzare questo protocollo la cui procedura è descritta nel link.

COMANDI PRINCIPALI GIT

cd pathFile → va nella cartella dove si vuole agire

I seguenti comandi vanno fatti nella directory

git clone _linkRepository → Fa un primo clone della repository

git status → Per verificare se la versione locale della repository è aggiornata rispetto quella online, o se occorre fare qualche push / pull

git add --all → per aggiungere tutte le modifiche

git commit -am "commento" → per aggiungere una descrizione delle modifiche fatte (è bene farlo, in inglese)

git push → per fare l'upload delle modifiche fatte

git pull → per fare il download delle modifiche di altri

git stash → per eliminare tutte le ultime modifiche fatte in locale delle quali ancora non si è fatto un push

Freccia su → ultimi comandi

Dunque un tipico esempio di lavoro è il seguente:

- **git status** → controllare se ci sono modifiche
- **git pull** → fare il download delle modifiche degli altri sviluppatori
- Lavorare sulle proprie classi java assicurandosi che nessuno lo stia facendo in contemporanea
- **git add --all**
- **git commit -am "commento"**
- **git push**

4. Setup di JPAD

Il lavoro in Eclipse

Benché sia ovviamente possibile scrivere un qualsiasi codice utilizzando un semplicissimo editor di testo, per la stesura di codici complessi si ricorre agli IDE. Un IDE è un software che, in fase di programmazione, aiuta i programmatori nello sviluppo del codice sorgente, comportandosi come un editor di testo ma con tutte le funzionalità per aiutare nella programmazione, come correggere errori nel codice, completare i comandi etc. L'IDE utilizzato nello sviluppo di jpad è Eclipse Oxygen, ma potrebbe essere ovviamente diverso per ogni programmatore. Eclipse è gratuito e scaricabile dal sito ufficiale

<http://www.eclipse.org/downloads/eclipse-packages/>

Si consiglia la versione Eclipse IDE for Java EE Developers, essendo la più completa. A questo punto occorre importare i progetti e settare tutte le variabili d'ambiente.

Importare i progetti:

- Tasto destro su Project Explorer → import → import. Figura 8,(1)
- Existing projects into workspace → next. Figura 8,(2)
- Browse → nella cartella dove è stata scaricata la repository. Figura 8,(3)
- Per le cartelle: JPADSandbox_v2, JPADCore_v2, JPADCommander, JPADConfig, cliccarci sopra, spuntare search for nested project → finish

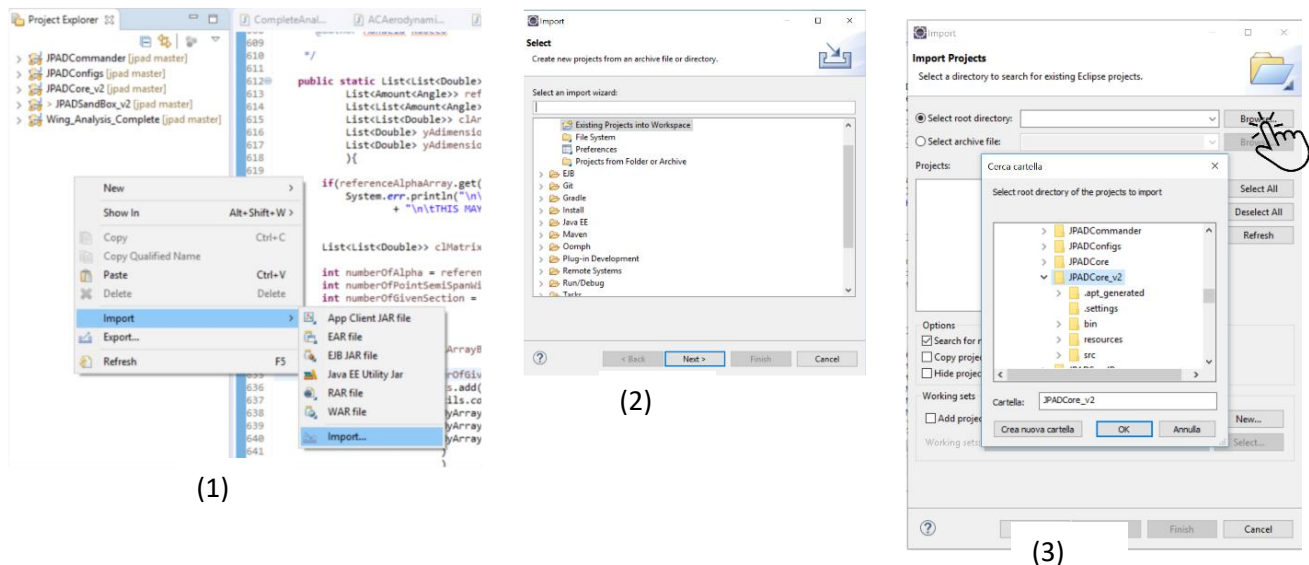


Figura 8: Come importare progetti in Eclipse.

Immettere le variabili d'ambiente:

In Eclipse vanno definite alcune variabili d'ambiente che sono variabili con uguale nome per ogni sviluppatore, ma che puntano al proprio percorso su ogni PC. Questo serve per poter puntare a librerie esterne (le quali, ovviamente, avranno un path differente su ogni PC), o ad altri file ausiliari.

- Dal menu principale windows → preferences → java → Build path → Classpath Variables
- New, e aggiungere le due variabili d'ambiente, Figura 9.
 - Name: LIBRARY_FOLDER
Path: Il percorso della cartella libraries di JPAD
 - Name: ECLIPSE_FOLDER
Path: Il percorso della cartella plugins in libraries di jpad

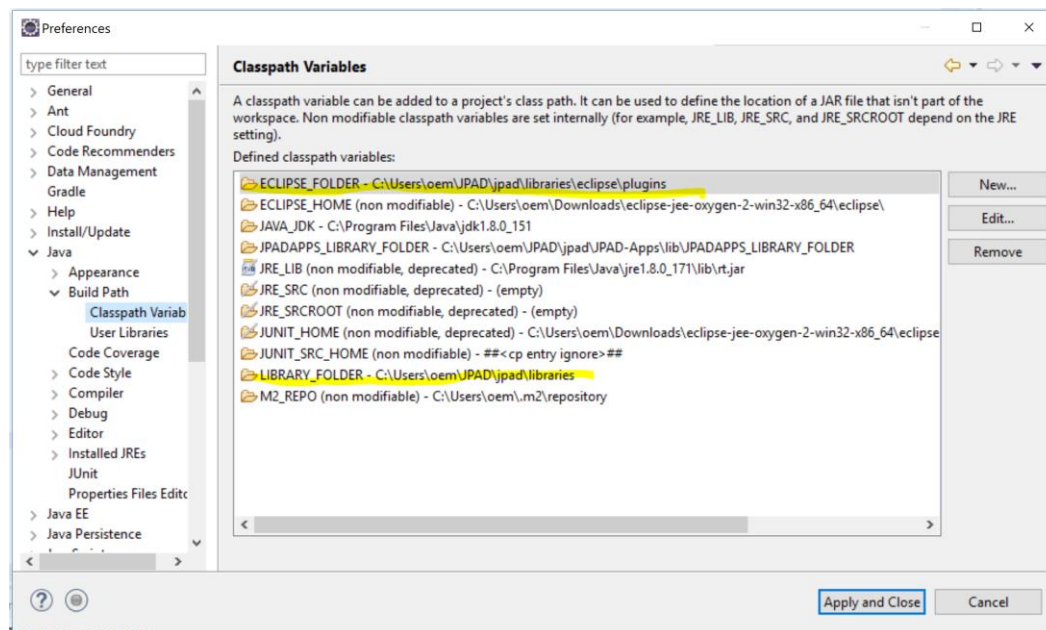


Figura 9: Variabili d'ambiente da inserire

Variabile di Run Configuration:

In un qualsiasi progetto, per poter fare un run e quindi testare alcune funzionalità, è necessario definire una **Run Configuration**, ossia la configurazione che permette di definire quale è la classe Main (quindi quale test si vuole fare), se ci sono eventuali input esterni ecc.

Le Run Configuration di java sono condivise in rete, quindi saranno già disponibili dopo il download del progetto. Per vedere quali sono le Run Configuration già disponibili e create da altri utenti, nel tab menu in alto Run → Run Configurations, Figura 10, Figura 11.

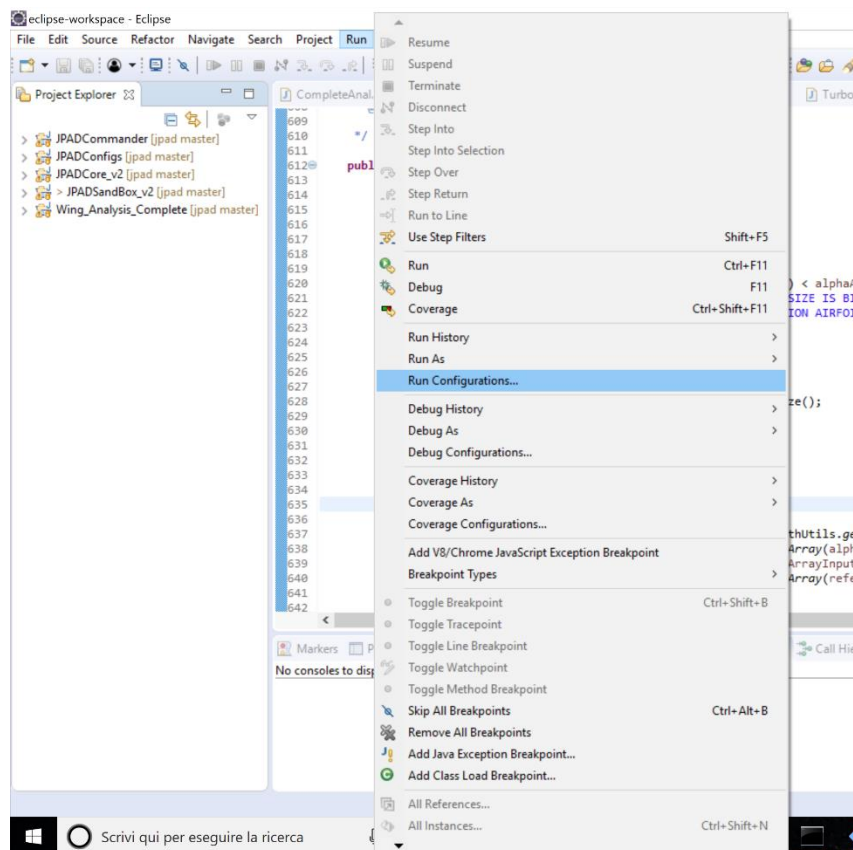


Figura 10: Run configurations

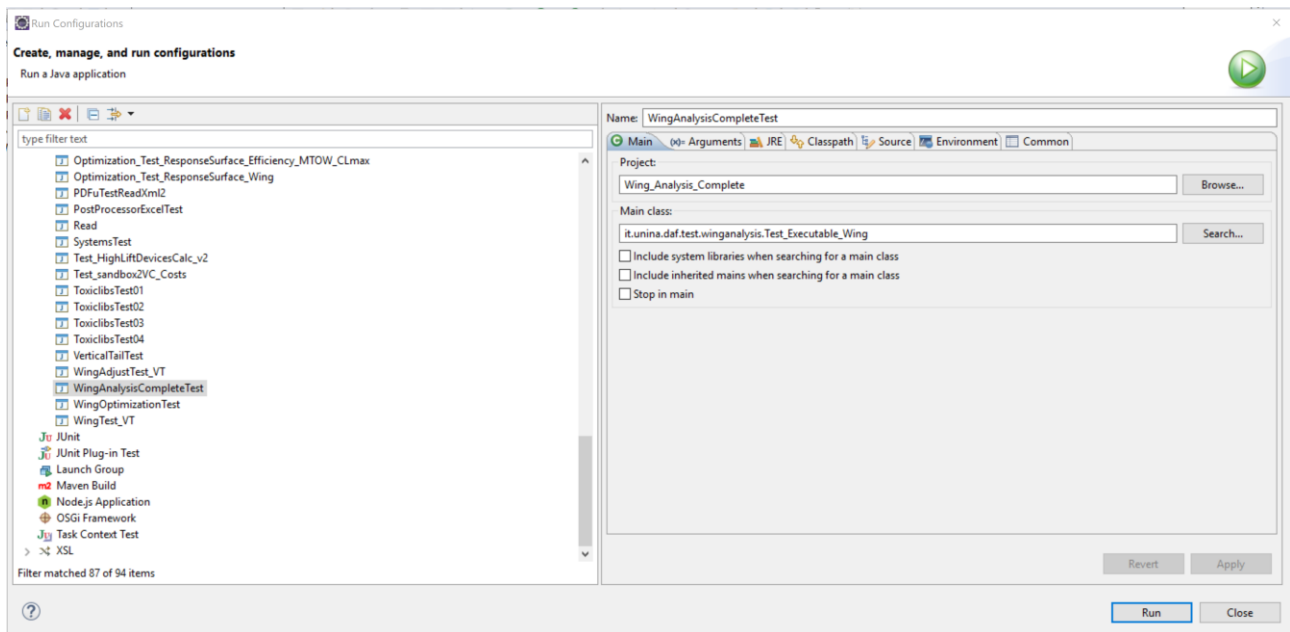


Figura 11: Esempio di Run Configuration

Per fare il run di una classe di test (classe Main), è possibile cliccare su Run di Figura 11, oppure cliccare **ctrl+F11** sulla classe Main.

Al primo utilizzo di JPAD, e solo al primo, è necessario inserire una variabile nelle Run configuration.

- Selezionare una Run configuration come visto in di Figura 11
- Dal tab menù, cliccare Environment, Figura 12, (1)
- Selezionare la variabile Path e cliccare su Edit, Figura 12, (2)(3)
- Cliccare su Variables → Edit Variables, Figura 12, (4)(5)
- Definire una nuova variabile con New:
Name: JPAD_ROOT
Value: il path della cartella jpad scaricata da GitHub

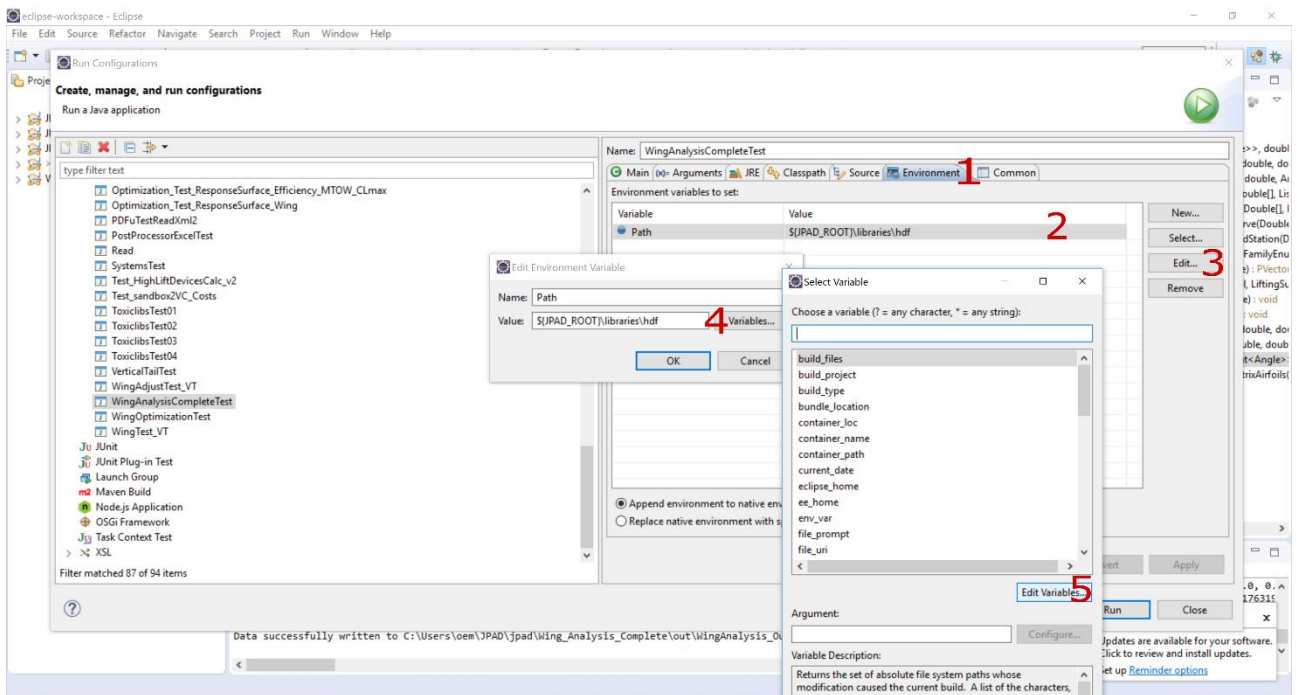


Figura 12: Come definire nuove variabili in Run Configuration.

A questo punto, per far funzionare correttamente la classe di test principale, occorre avere i database motore che a causa della presenza di dati sensibili, da non diffondere, saranno passati su penna usb.

Lavoro nella sandbox:

Per iniziare a lavorare in jpad è bene creare una propria cartella ove saranno messi tutti i progetti. Tali cartelle sono chiamate Sandbox, e sono nel progetto JPADSandbox_2. Per fare ciò

- Cliccare sulla freccia laterale JPADSandbox_v2 per aprirne il contenuto
- Cliccare su src
- Tasto destro su sandbox2 → new → Package
- Creare un nuovo package con le proprie iniziali di nome e cognome. Per metterlo all'interno di sandbox2, occorre mettere il nome come segue: sandbox2.nc, come visto in Figura 13.

La sandbox ha già le dipendenze con gli altri package, quindi è possibile utilizzare le funzionalità già presenti in jpad, creando una classe Main o altre funzionalità nella propria sandbox.

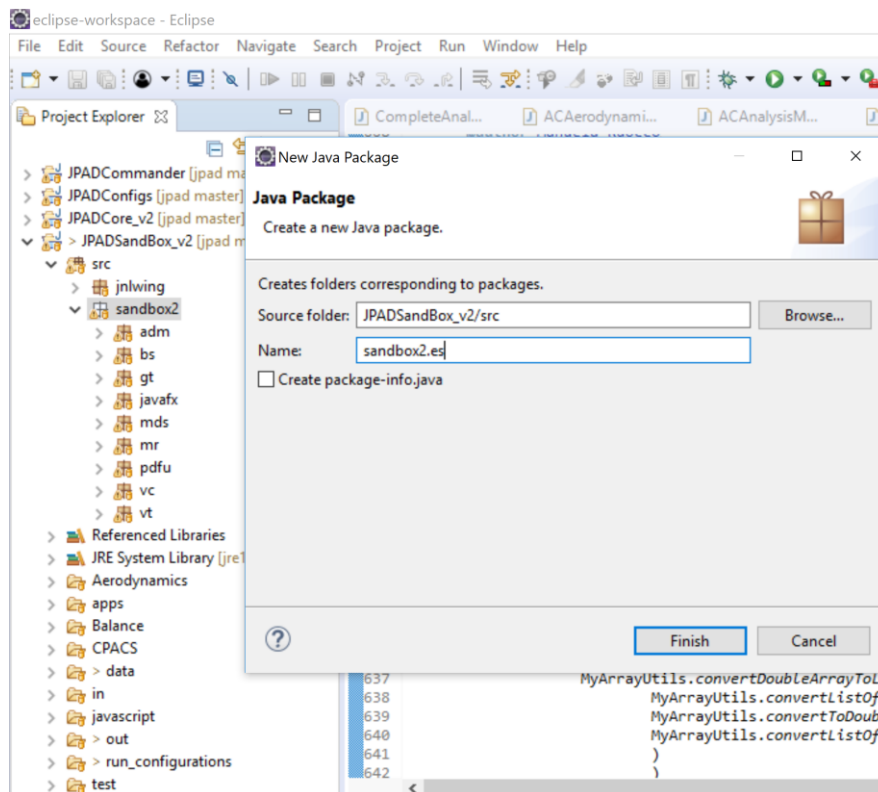


Figura 13: Creazione propria sandbox

5. Gestione delle GUI

Una GUI (Graphical User Interface), è un'interfaccia grafica che ha lo scopo di aiutare l'utente nell'utilizzo del programma. Ci sono diversi modi per implementare una GUI, in jpad questa è in fase di sviluppo tramite la libreria Java FX e Scene Builder, che aiuta nella creazione di GUI.

Per configurare Java FX seguire la seguente procedura: <https://www.eclipse.org/efclipse/install.html>

Per scaricare Scene Builder : <http://gluonhq.com/products/scene-builder/#download>

Per imparare a sviluppare GUI in java la seguente playlist può dare un notevole contributo: <https://www.youtube.com/playlist?list=PLpFneQZCNR2ktqseX11XRbC5Kyzdg2fbo>

6. Riepilogo step per inizio lavoro su JPAD

Di seguito è riportato un riepilogo di tutti i passaggi necessari, descritti in questo documento:

1. Scaricare e Installare Java 8, jdk e jre, immetterli nella variabile d'ambiente Path.
2. Creazione account su GIT HUB
3. Download della Bash di Git
4. Generazione e aggiunta chiave protocollo SSH
5. Clone della repository online
6. (Eventuale) Configurazione LFS
7. Download Eclipse
8. Importazione progetti
9. Variabili d'ambiente
10. Variabile di Run Configuration
11. Database motori