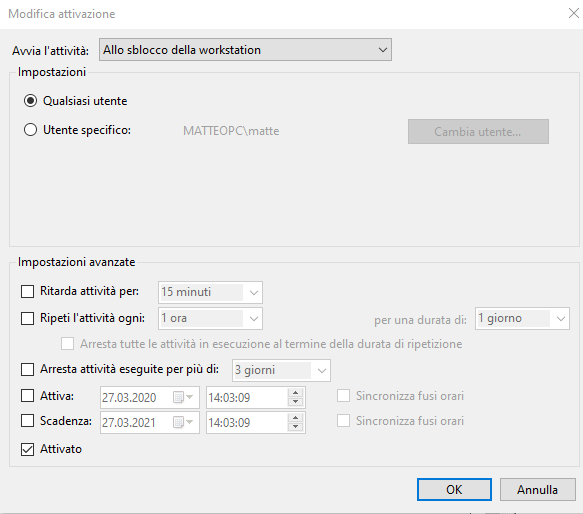
# Lavori svolti

|  |  |  |  |
| --- | --- | --- | --- |
| Orario | lavoro svolto | persone | ID Lavoro |
| 8:20 – 11:35 | Stesura del Gantt | Matteo | 1 |
| 8:20 – 14:45 | Script che si avvia al login (Windows) | Bruno, Matteo (dalle 13:15) | 2 |
| 15:00 – 16:30 | Script per importare l’xml | Bruno, Matteo | 3 |
| 8:20 – 9:55 | Lettura di un file CSV (Java) | Luca | 4 |
| 10:05 – 11:35 | Esecuzione al login (macOS) | Luca | 5 |
| 13:15 – 16:30 | Pacchetto installazione (macOS) | Luca | 6 |

# Lavori

**Nota:** Il giorno 20/03/2020 è stato perso a causa di un ponte imposto dal cantone (causa coronavirus). Nonostante questo inconveniente pensiamo di starci nei tempi, grazie a molte lezioni estremamente produttive.

1. Ho dovuto rifare il Gantt a causa di un cambio di software per gestirlo, questo cambio è stato forzato a causa del coronavirus (sono passato da Microsoft Project a ProjectLibre). Purtroppo ProjectLibre non riesce ad interpretare al meglio i file di Microsoft Project e questo ha incasinato il Gantt, per risolverlo ho deciso di rifare completamente il Gantt (basandomi comunque su quello che già avevamo ma era sbagliato in termini di date). Grazie a questo inconveniente sono riuscito a fare un Gantt estremamente più preciso e fedele.
2. Ci siamo accorti che lo script partiva unicamente al primo avvio del PC e non partiva se veniva fatta una disconnessione e poi di nuovo l’accesso. Per aggirare questo problema siamo tornati nel Task Scheduler, nelle proprietà del Task (chiamato FaceLock) e abbiamo impostato l’avvio del task all’accensione della workstation. Per mettere questo evento bisogna andare nel tab attivazione, fare modifica e scegliere all’avvio della workstation. Fatto questo ho esportato il task come xml e abbiamo cercato un modo per importarlo tramite linea di comando cambiando l’utente che eseguiva il task (perché cambiando PC l’utente che era dedicato all’avvio del task doveva essere differente). I comando è il seguente:  
     
   I problemi adesso sono due, ovvero il fatto di eseguire il cmd come amministratore per riuscire a completare l’operazione e l’inserimento della password dell’utente.
3. Grazie ad un file bat siamo riusciti a scrivere uno script che permetta di importare automaticamente il file xml contenente le informazioni del task. Questo script verrà eseguito ogni volta che si premerà applica nell’interfaccia grafica per impostare le impostazioni e, una volta importato il file xml, lo eliminerà in modo che così la prossima volta saprà che lo ha già importato. L’unica cosa brutta è che lo script richiederà all’utente i permessi di amministratore e di inserire la propria password (altrimenti non importerà lo script).
4. È stato scritto un metodo che si occupa di leggere il file CSV delle impostazioni. Ciò è stato fatto usando la classe FileReader della libreria base di Java.

A screenshot of a cell phone

Description automatically generated

1. Ho passato all’incirca un’ora nel cercare un metodo valido per poter eseguire il pacchetto .app del programma al login dell’utente.

Sono stati testati molteplici metodi, tra i quali la scrittura di un file .plist, l’esecuzione di un comando AppleScript. Quest’ultima è risultata la più comoda e funzionante, è quindi stata applicata.

Questo comando è stato applicato nel programma d’impostazioni, al selezionamento di una casella selezionabile. Questa permette di aggiungere e rimuovere un’applicazione dai *Login Items* del sistema operativo.A screenshot of text

Description automatically generated

I due comandi sono stati salvati in due script che vengono eseguiti separatamente, rispettivamente quando viene attivata o disattivata l’esecuzione al login.

Di seguito il codice dei suddetti script.





1. Ho iniziato a sviluppare il pacchetto d’installazione per macOS. Ciò però non è stato portato a termine.

Lo strumento utilizzato è *Platypus*.

È stata creata una cartella contenente le risorse (script e programmi) e l’applicazione (.app), creata con *Platypus*.

# Problemi riscontrati

1. È stato impiegato molto tempo nel trovare il modo con il quale fosse il comando funzionante, che non fosse ancora stato deprecato.
2. Lo script, eseguito tramite l’applicazione generata da *Platypus* non riesce ad accedere alle risorse

# Programma di massima per la prossima giornata di lavoro

* Implementare lo script dell’importazione dell’xml nell’interfacci agrafica (per windows).
* Terminare il pacchetto d’installazione macOS.

# Punto rispetto alla pianifica

Estremamente avanti.