

# GESTIONE CASINÒ | Diario di lavoro - 22.03.2019

---

Matan Davidi, Thor Döblin, Matteo Forni, Carlo Pezzotti, Mattia Toscanelli

Trevano, 22 marzo 2019

## Lavori svolti

---

È stato deciso che non verrà installato un mail server sul server remoto di produzione perché risulta più semplice utilizzare la libreria "PHPMailer" [PHPMailer](https://github.com/PHPMailer/PHPMailer) (<https://github.com/PHPMailer/PHPMailer>) utilizzando come server di posta elettronica Hotmail tramite l'indirizzo email "gruppoCasin02018@hotmail.com".

Matan oggi ha ripreso la documentazione, scrivendo l'abstract all'interno del capitolo 1.2, Abstract.

Mattia Toscanelli ha cercato delle immagini da aggiungere al sito, senza diritti d'autore. Il sito nel quale le ha trovate è il seguente:  
<https://www.pexels.com/search/casino/>

Oggi Carlo e Mattia hanno lavorato insieme per mettere insieme al meglio la part front-end e back-end. Siamo riusciti con successo a fare ciò. Carlo aveva bisogno di sapere come fosse la struttura per gestire gli errori e Mattia mi ha fatto notare che nel progetto esiste una libreria che permette di notificare. Alla fine siamo giunti alla conclusione che è meglio utilizzare questo approccio.

Per passare il testo di errore alla notifica uso lo stesso approccio descritto nel diario scorso, ovvero di salvare l'errore su un cookie e mostrarlo in seguito, grazie al seguente metodo:

```
function getCookie(cname) {
    console.log("Loo");
    var name = cname + "=";
    var decodedCookie = decodeURIComponent(document.cookie);
    var ca = decodedCookie.split(';');
    for(var i = 0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) == ' ') {
            c = c.substring(1);
        }
        if (c.indexOf(name) == 0) {
            var returnValue = c.substring(name.length, c.length);
            while(returnValue.includes("+")){
                returnValue = returnValue.replace("+", " ");
            }
            console.log(returnValue);
            return returnValue;
        }
    }
    return "";
}

if(getCookie("error") != ""){
    var inputs = document.getElementsByTagName("input");
    inputs[0].style.backgroundColor = "#ffcccc";
    inputs[1].style.backgroundColor = "#ffcccc";
    $.notify(getCookie("error"), { position:"bottom left" });
}else{
    inputs[0].style.backgroundColor = "white";
    inputs[1].style.backgroundColor = "white";
}
```

La seguente funzione in base al nome del cookie ritorna il suo valore.

La vita del cookie è molto breve, infatti abbiamo deciso di farlo vivere solo 1 secondo.

Oggi Matteo ha risolto tutti i problemi di permessi che l'utente di Jenkins riscontrava con tutte le cartelle di git ed i relativi files di configurazione.

Ha poi inserito lo script come Post Build Task nel lavoro di Jenkins ed ha eseguito dei test di funzionamento che non hanno ritornato risultati positivi.

Oggi Thor ha continuato la creazione del codice Java del WebDriver di Selenium, che testa la navigazione tra le pagine di Registrazione, Login e Smarrimento password, il problema sulla quale si è concentrato oggi è il come gestire l'attesa tra una pagina e l'altra prima di eseguire una qualsiasi iterazione con essa. Per questo sono stati provati vari metodi che non hanno prodotto un risultato positivo.

## Problemi riscontrati e soluzioni adottate

---

Matteo ha riscontrato inizialmente molti problemi di permessi che sono stati tutti risolti cambiando proprietario o permessi della cartella o file in questione. Dopodiché inserendo lo script come Post Build Task di Jenkins ha notato che esso non viene mai eseguito perché vi è sempre un messaggio che dice **Logical operation result is FALSE** anche se tutto viene eseguito senza alcun problema.

```

OK (7 tests, 7 assertions)
Performing Post build task...
Could not match :build : False
Logical operation result is FALSE
Skipping script : cd /var/lib/jenkins/workspace
cd GestioneCasino
git pull
cd ..
rsync -a --exclude=".*" ./GestioneCasino/. ./GestioneCasinoProduction/
cd ./GestioneCasinoProduction
git remote set-url origin https://github.com/samtcasino/GestioneCasinoProduction.git
git pull origin master --allow-unrelated-histories
git add .
git commit -m "Jenkins commit"
git push

END OF POST BUILD TASK : 0
Finished: SUCCESS

```

Thor ha provato, insieme all'aiuto di Matan un paio di metodi per gestire l'attesa di caricamento tra una pagina e l'altra, i seguenti metodi dovrebbero attendere che la pagina si sia caricata completamente prima di eseguire una qualsiasi altra operazione su di essa.

Con il seguente metodo si attende che lo stato della pagina ritorni "complete", e che quindi la pagina si sia caricata completamente, tuttavia sembra che la pagina ritorni lo stato "complete" quando in realtà si trova ancora in una fase di caricamento.

```

ct.driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
new WebDriverWait(ct.driver, 10).until(
    driver -> ((JavascriptExecutor)driver).executeScript("return document.readyState").equals("complete"));

```

Anche in questo caso si aspetta che l'elemento il link Accedi sia accessibile prima di eseguire una qualsiasi azione, tuttavia per qualche motivo anche qui sembra che il link sia cliccabile quando in realtà non risulta così.

```

''' Java
WebDriverWait wait = new WebDriverWait(ct.driver, 60);
wait.until(ExpectedConditions.elementToBeClickable(By.linkText("Accedi")));

```

Con il seguente metodo ci si aspetta che il titolo della pagina sia uguale a quello che ci si aspetta, tuttavia quando la pagina sta ancora caricando il t



```

Java
(new WebDriverWait(ct.driver, 10)).until(new ExpectedCondition() {
    public Boolean apply(WebDriver d) {
        return d.getTitle().startsWith("gestionecasino - login");
    }
});

```

Questi metodi ritornano tutti l'errore:

unknown error: Element ... is not clickable at point (781, 43). Other element would receive the click:

...

...

Che indica che si sta provando ad eseguire il comando per cliccare al quando in verità la pagina non si è ancora caricata.

## Punto della situazione rispetto alla pianificazione

Rispetto alla pianificazione siamo in orario.

## Programma di massima per la prossima giornata di lavoro

Matteo ha intenzione di far sì che Jenkins esegua lo script.

Thor per la prossima giornata di lavoro ha intenzione di concludere i test che riguardano la navigazione tra pagine e cercare un framework per i protocolli di test.