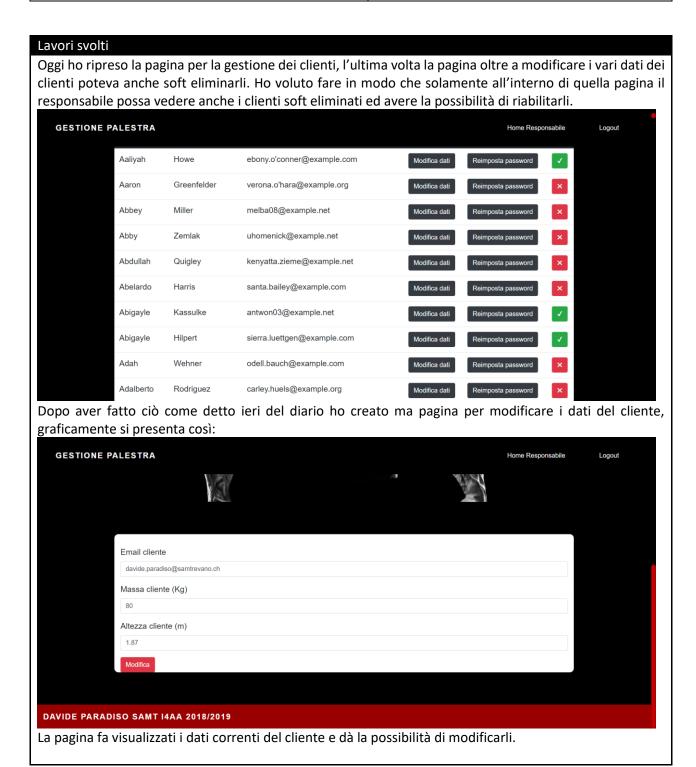
## Diario di lavoro

| Luogo | Scuola Arti e Mestieri Trevano |
|-------|--------------------------------|
| Data  | mercoledì, 30 gennaio 2019     |



Nome Progetto: Gestione Palestra

```
Per far rappresentare i dati (quindi la grafica della pagina in generale) ho fatto una funzione nel controller:
function modificaDati($id) {
    // Riporto l'id del cliente nella pagina
    // Richiamo la classe del responsabile
    require once 'application/models/responsabile.php';
    // Creo l'oggetto responsabile
    $r = new Responsabile();
    // Mi faccio restituire la mail del cliente dal responsabile
    $mailCliente = $r->getClienteMail($id);
    // Richiamo la classe cliente
   require once 'application/models/cliente.php';
    // Creo l'oggetto cliente
    $c = new Cliente($mailCliente);
    // Prendo i dati del cliente
    $pesoCliente = $c->getMassa();
    $altezzaCliente = $c->getAltezza();
    // View
    require 'application/views/ template/header.php';
    require 'application/views/responsabile/modificaDati.php';
    require 'application/views/ template/footer.php';
Per fare la modifica dei dati del cliente invece ho creato un'altra funzione sempre nel controller:
function changeDataCliente($id) {
    if ($ SERVER["REQUEST METHOD"] == "POST") {
        // Controllo che i campi non siano vuoti
        $mail = isset($ POST['mail']) ? $ POST['mail'] : null;
        $peso = isset($ POST['peso']) ? $ POST['peso'] : null;
        $altezza = isset($ POST['altezza']) ? $ POST['altezza'] : null;
        // Se anche un campo é vuoto ritorno alla home del responsabile
        if ($mail == null || $peso == null || $altezza == null)
            header("location: " . URL . "clienteController/responsabile");
        // Richiamo la classe cliente
        require once 'application/models/cliente.php';
        // Calcolo il BMI
        $bmi = Cliente::calcBMI($altezza, $peso);
        // Richiamo la classe del responsabile
        require once 'application/models/responsabile.php';
        // Creo l'oggetto responsabile
        $r = new Responsabile();
        // Aggiorno i nuovi dati del cliente
        $r->updateDataCliente($id, $altezza, $peso, $BMI, $mail);
        // Ritorno alla pagina di modifica
        header("location: " . URL . "responsabileController/listaModifiche");
```

La funzione che effettivamente modifica i dati del cliente è **updateDataCliente()** che viene richiamata tramite un oggetto di tipo **Responsabile**:

```
public function updateDataCliente($id, $altezza, $peso, $BMI, $mail) {
    $conn = $this->db->getConnection();
    // Faccio le query
    $query = "UPDATE cliente SET Altezza = $altezza WHERE id = $id";
    $conn->query($query);
    $query = "UPDATE cliente SET Massa = $peso WHERE id = $id";
    $conn->query($query);
    $query = "UPDATE cliente SET BMI = $BMI WHERE id = $id";
    $conn->query($query);
    $query = "UPDATE cliente SET BMI = '$mail' WHERE id = $id";
    $conn->query($query);
    $conn->query($query);
    $conn->close();
}
```

Dopo aver fatto questa parte di sito ho cominciato a riempire il database collegando i vari attrezzi ai parametri.

| 🤌 id      | Parametro   | id_attrezzo |
|-----------|-------------|-------------|
| 18        | Ripetizioni | 6           |
| 19        | Serie       | 6           |
| 20        | Carico      | 7           |
| 21        | Ripetizioni | 7           |
| 22        | Serie       | 7           |
| <b>23</b> | Carico      | 8           |
| 24        | Ripetizioni | 8           |
| <b>25</b> | Serie       | 8           |
| <b>26</b> | Carico      | 9           |
| 27        | Ripetizioni | 9           |
| 28        | Serie       | 9           |

## Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In orario con la mia pianificazione.

## Programma di massima per la prossima giornata di lavoro

Iniziare l'implementazione della pagina per creare gli esercizi.