

*Team di Sviluppo:*

Cociug Raul Andrei - Sviluppatore

Madiotto Gabriel - Sviluppatore

Azienda: DomotiX

# Documentazione Tecnica

Versione	Data	Autore	Clienti
1.0	12/05/2025	Cociug, Madiotto	Tollot, Rossi

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Premessa</b>	<b>2</b>
<b>3</b>	<b>Obiettivi del Progetto</b>	<b>2</b>
<b>4</b>	<b>Struttura del Progetto</b>	<b>2</b>
<b>5</b>	<b>Architettura Tecnica</b>	<b>2</b>
5.1	Tecnologie Utilizzate . . . . .	2
5.2	Componenti Principali . . . . .	3
<b>6</b>	<b>Gestione Stato e Interazione</b>	<b>3</b>
6.1	Esempio - Gestione Luci . . . . .	3
6.2	Modalità Giorno/Notte . . . . .	3
6.3	Gestione Cookie . . . . .	3
<b>7</b>	<b>Funzionalità Implementate</b>	<b>4</b>
<b>8</b>	<b>Responsive Design</b>	<b>4</b>
<b>9</b>	<b>Organizzazione del Codice</b>	<b>4</b>
<b>10</b>	<b>Possibili Estensioni Future</b>	<b>5</b>
<b>11</b>	<b>Note per Sviluppatori</b>	<b>5</b>
<b>12</b>	<b>Debug e Manutenzione</b>	<b>5</b>
<b>13</b>	<b>Conclusioni</b>	<b>6</b>

# 1 Introduzione

Il seguente documento fornisce una descrizione tecnica dettagliata dell'interfaccia grafica della casa domotica. L'interfaccia consente agli utenti di interagire con gli elementi di un'abitazione.

## 2 Premessa

La configurazione e la gestione della rete non fanno parte dei servizi offerti dal sistema. È responsabilità dell'utente disporre di un ambiente di rete compatibile e funzionante. Le eventuali manutenzioni o modifiche da apportare sono anch'esse al di fuori dei servizi offerti dall'azienda. Una volta consegnato il prodotto finito al cliente l'azienda non avrà più alcuna responsabilità sul progetto e il cliente dovrà riconoscere il diritto d'autore dell'azienda sul progetto.

## 3 Obiettivi del Progetto

- Disporre graficamente un ambiente domotico tramite interfaccia grafica web;
- Permettere l'interazione dell'utente con i bottoni e la mappa della casa;
- Realizzare un'interfaccia responsive, compatibile con diversi dispositivi;
- Gestire lo stato dei dispositivi mediante codice JavaScript locale.

## 4 Struttura del Progetto

- `index.html` – struttura principale della pagina;
- `style.css` – gestione grafica e layout delle pagine;
- `buttons.css` – gestione grafica e layout dei bottoni;
- `img-casa.css` – gestione grafica e layout della mappa della casa;
- `script.js` – logica, interazioni e simulazioni dinamiche;
- `/img/` – directory contenente le immagini (icone, mappa, sfondi);
- `/Stanze/` – directory contenente i file HTML delle stanze della casa.

## 5 Architettura Tecnica

### 5.1 Tecnologie Utilizzate

- **HTML5**: per la struttura semantica e visiva dell'interfaccia;
- **CSS3**: per la formattazione, animazioni e supporto responsive;
- **JavaScript Vanilla**: per la logica client-side e l'interazione DOM;
- **Cookie API**: per la simulazione di stato persistente lato client.

## 5.2 Componenti Principali

- **Mappa interattiva:** ogni stanza o area è cliccabile e mostra controlli dedicati;
- **Moduli funzionali:** luci, porte, meteo, temperatura, Roomba, irrigazione, ora, velocità vento, wattaggio della casa, condizionatore, termosifoni, serrande;
- **Pannello superiore:** titolo della stanza corrente, modalità giorno/notte automatica o forzata;
- **Script dinamico:** ogni modulo ha funzioni dedicate con listener su eventi.

## 6 Gestione Stato e Interazione

Ogni elemento cliccabile (es. bottoni luce, porta) è associato a una funzione JavaScript che aggiorna dinamicamente lo stato, applica modifiche al DOM e, dove previsto, salva la preferenza tramite cookie.

### 6.1 Esempio - Gestione Luci

```
// Spegnimento luci
for (let i=0; i < luci.length; i++) {
    luci[i].src = "img/lamp-spenta.png"
}
deleteCookie("btn-luci");

// Accensione luci
for (let i=0; i < luci.length; i++) {
    luci[i].src = "img/lampadina.png";
}
setCookie("btn-luci", "true", 1);
```

### 6.2 Modalità Giorno/Notte

L'interfaccia cambia automaticamente colore in base all'orario (06:00-18:00 giorno, altrimenti notte). L'utente può forzare manualmente la modalità tramite un pulsante.

### 6.3 Gestione Cookie

```
// Settaggio di un Cookie
function setCookie(name, value, days) {
    const date = new Date();
    date.setTime(date.getTime() + days * 1000 * 60 * 60 * 24);
    let expires = "expires=" + date.toUTCString();
    document.cookie = `${name}=${value}; ${expires}; path=./`;
}
```

Il parametro della funzione "setCookie" name identificherà il nome del Cookie che si vuole creare, value il valore associato a quel nome e days l'ammontare di giorni che si desidera mantenere attivo il Cookie.

Formato di un Cookie in visualizzazione: "name=value".

Formato di un Cookie in creazione: "name=value; expires=<UTC Date>; path=./".

```
// Prendere valore di un Cookie
function getCookie(name) {
    const cookieArray = document.cookie.split("; ");

    let result = null;
    cookieArray.forEach(element => {
        if(element.indexOf(name) == 0) {
            result = element.substring(name.length + 1);
        }
    });
    return result;
}
```

Cicla tutto l'array dei Cookie, precedentemente inizializzato, e se trova un elemento dove il 'name' passato come parametro si trova all'inizio di esso (`indexOf(name) = 0`) allora ritorna il valore di quel Cookie.

```
// Eliminazione Cookie
function deleteCookie(name) {
    setCookie(name, null, null)
}
```

## 7 Funzionalità Implementate

- Accensione/spegnimento luci (per stanza o globalmente);
- Apertura/chiusura porte;
- Controllo Roomba (ON/OFF);
- Attivazione irrigazione;
- Visualizzazione meteo (con temperatura, vento);
- Visualizzazione orario attuale;
- Modalità responsive adattiva.

## 8 Responsive Design

L'interfaccia è stata progettata per adattarsi automaticamente a smartphone, tablet e desktop. L'utilizzo del tag `meta viewport` e delle `media queries` CSS permette la fluidità dei contenuti.

## 9 Organizzazione del Codice

### HTML

Layout semplice con tre parti principali (header, mappa della casa, pannello) basato su griglia CSS, con elementi identificabili via ID e classi. Gli elementi principali sono contenuti in div con attributi semantici.

## CSS

Lo stile principale degli elementi HTML si trova nel file `style.css`, per lo stile dei bottoni si ha `buttons.css` e infine per la mappa della casa si ha `img-casa.css`. Per lo stile dinamico e semplificazione di codice si hanno variabili che contengono colori per i vari elementi, questo permette di avere uno stile dinamico con un tema chiaro e un tema scuro.

## JavaScript

Lo script è suddiviso in blocchi per ogni modulo (luci, porte, meteo, ecc.). Ogni blocco contiene:

- Funzione `toggle` o `update`;
- Event Listener al caricamento;
- Salvataggio dello stato con `cookie`.

## 10 Possibili Estensioni Future

- Integrazione con server back-end e database;
- Autenticazione utenti;
- Storico degli eventi;
- Collegamento a dispositivi reali (es. MQTT, Arduino, ESP32).

## 11 Note per Sviluppatori

- Utilizzare nomi coerenti per ID e classi;
- Usare la stessa formattazione per la modifica o aggiunta di altri parametri o bottoni;
- Testare sempre su dispositivi mobili e desktop;
- Separare la logica in moduli riutilizzabili.

## 12 Debug e Manutenzione

Problema	Causa	Soluzione
Evento non funzionante	ID errato o listener mancante	Controllare ID elemento e listener JS
Luci non si accendono	Immagini assenti o JS corrotto	Verificare classi nel DOM e/o eventuali errori nel JS
Stato non salvato	Cookie disattivati o script errato	Verificare cookie nel browser
UI non responsive	Media query non applicata correttamente	Controllare larghezza dispositivo e/o breakpoints

## 13 Conclusioni

L'interfaccia della casa domotica sviluppata dalla DomotiX rappresenta un ambiente modulare e ben strutturato, pronto ad essere esteso e integrato. Questa documentazione tecnica è concepita per agevolare futuri sviluppatori nella comprensione, manutenzione e miglioramento del sistema.

*Per ulteriori chiarimenti, fare riferimento ai commenti all'interno dei file sorgenti.*