

Specifiche tecniche e best practice per lo sviluppo del Sito Web

Sommario

1.	Linee guida per lo sviluppo responsive dei siti	3
1.1.	Ottimizzazione del testo per la lettura.....	3
1.2.	Impostazione del viewport.....	3
1.3.	Dimensionamento dei contenuti in base al viewport	4
1.4.	Utilizzo delle media query CSS	4
1.5.	Utilizzo di unità relative.....	4
1.6.	Layout “Column drop”	4
1.7.	Immagini responsive	5
2.	SEO	5
2.1.	Mobile-First Approach.....	6
2.2.	Utilizzo dei Breadcrumbs.....	6
2.3.	Sitemap e Robots.txt	6
2.4.	URLs	7
2.5.	Internalizzazione	7
2.6.	HTTPs.....	7
2.7.	Pagina 404	7
2.8.	Perfomance e pagespeed.....	7
2.9.	Headings	8
2.10.	Immagini	8

Il presente documento ha lo scopo di indicare le specifiche e le best practice da seguire per le attività di sviluppo del Sito Web.

La mancata aderenza ai principi elencati per ogni ambito può comportare la richiesta da parte di ASPI al rifacimento in garanzia, a carico del SI, di quanto sviluppato.

1. Linee guida per lo sviluppo responsive dei siti

L'implementazione di FrontEnd dei siti dovrà prevedere un approccio Mobile-first e l'utilizzo delle tecniche di sviluppo web responsive.

Progettare i contenuti in funzione degli schermi di piccole dimensioni, quindi espandere lo schermo fino al punto in cui occorre inserire un breakpoint.

L'implementazione lato CSS dovrà comunque almeno prevedere i seguenti breakpoints:

- Desktop/Tablet:
 - Screen size $\geq 1024\text{px} \times 768\text{px}$
- Mobile:
 - Screen size $< 1024\text{px} \times 768\text{px}$

1.1. Ottimizzazione del testo per la lettura

Secondo la teoria classica della leggibilità, una colonna dovrebbe contenere da 70 a 80 caratteri per linea (circa 8-10 parole in italiano). Pertanto, quando un blocco di testo supera le 10 parole occorre inserire un breakpoint.

1.2. Impostazione del viewport

Le pagine ottimizzate per dispositivi multipli devono contenere un elemento meta viewport nella sezione 'head' del documento, che indica al browser come controllare dimensioni e scala della pagina. In particolare:

- Utilizzare i meta viewport tag per controllare larghezza e scala dei viewport dei browser.
- Inserire `width=device-width` per ottenere una corrispondenza con la larghezza dello schermo in pixel indipendenti dal dispositivo.
- Inserire `initial-scale=1` per stabilire una relazione 1:1 fra i pixel del CSS e quelli indipendenti dal dispositivo.
- Controllare l'accessibilità della pagina senza disabilitare l'opzione di scalabilità dell'utente.

1.3. Dimensionamento dei contenuti in base al viewport

A causa della variabilità di dimensioni e larghezza dello schermo in pixel CSS fra dispositivi diversi (es. telefoni, tablet o persino modelli diversi di telefono), la resa ottimale dei contenuti non deve basarsi su un viewport specifico:

- Non utilizzare elementi di grandi dimensioni e larghezza fissa.
- La resa ottimale dei contenuti non deve essere legata alla larghezza di un viewport specifico.
- Utilizzare i media query CSS per applicare diversi stili per gli schermi ampi e ristretti.

1.4. Utilizzo delle media query CSS

- Usare le media query per applicare gli stili in base alle caratteristiche del dispositivo.
- Utilizza min-width al posto di min-device-width per ottenere un'esperienza adatta al maggior numero di dispositivi possibile.
- Usare dimensioni relative degli elementi per evitare interruzioni della disposizione.

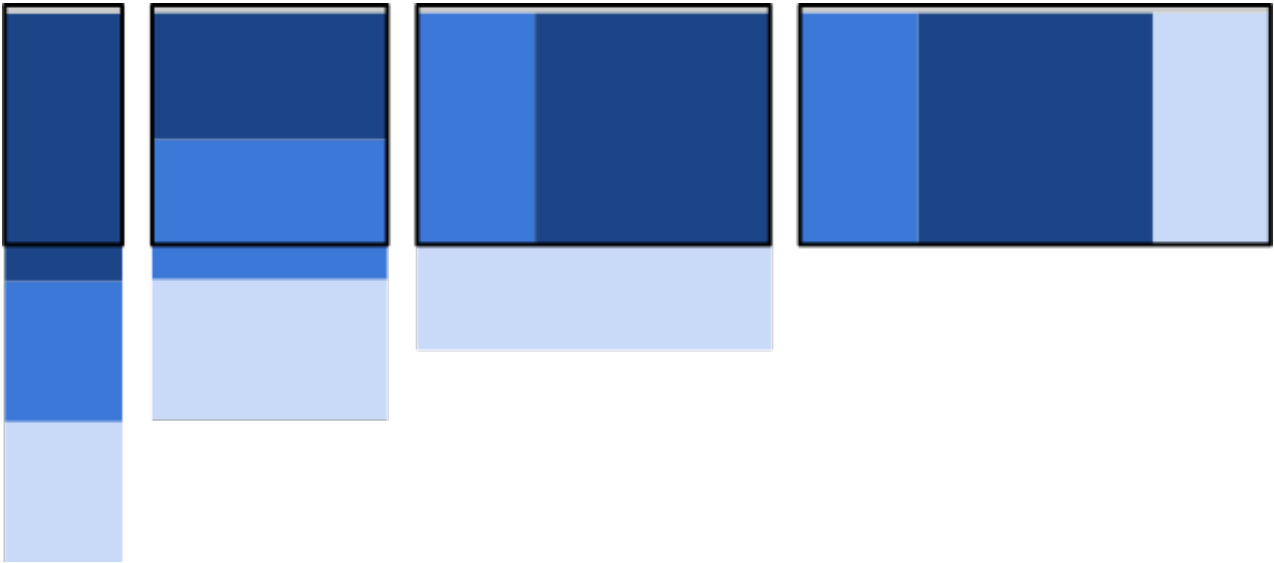
1.5. Utilizzo di unità relative

Due importanti concetti di design reattivo sono la fluidità e la proporzionalità, condizioni opposte alle disposizioni a larghezza fissa. L'utilizzo di unità di misura relative semplifica la creazione delle disposizioni ed evita la creazione involontaria di componenti troppo grandi per il viewport.

1.6. Layout “Column drop”

Utilizzare un layout “Column drop” per cui per i layout a più colonne a tutta larghezza, il rilascio della colonna impila semplicemente le colonne verticalmente poiché la larghezza della finestra diventa troppo stretta per il contenuto.

Alla fine, ciò comporta che tutte le colonne vengano impilate verticalmente. La scelta dei punti di interruzione per questo modello di layout dipende dal contenuto e dalle modifiche per ciascun disegno.



1.7. Immagini responsive

Il Web design responsive consente di modificare sia i layout che i contenuti in base alle caratteristiche del dispositivo. Per questo si richiede di:

- Utilizzare le dimensioni relative delle immagini per evitare l'overflow involontario del contenitore.
- Utilizzare l'elemento picture per specificare immagini diverse in base alle caratteristiche del dispositivo (operazione detta 'direzione artistica').
- Utilizzare srcset e il descrittore x nell'elemento img per indicare al browser l'immagine da utilizzare in presenza di diverse densità.

2. SEO

Al fine di garantire una corretta indicizzazione sui motori di ricerca, la società aggiudicatrice dovrà prevedere nello sviluppo tutte le azioni necessarie per ottemperare alle seguenti disposizioni in ambito SEO.

Lo sviluppo dovrà seguire le Best practice **SEO** (acronimo inglese per Search Engine Optimization che definisce tutte le attività di ottimizzazione di un sito web volte a migliorarne il posizionamento nei risultati organici dei motori di ricerca come Google e Bing) indicate nella "Guida introduttiva all'ottimizzazione per i motori di ricerca (SEO)" al seguente link: <https://support.google.com/webmasters/answer/7451184>.

Di seguito sono elencate alcune delle attività di ottimizzazione da seguire.

2.1. Mobile-First Approach

Secondo le regole di design rilasciate da Google, i siti devono essere sviluppati con un design responsive in modo che il server adatti il layout in base alla dimensione dello schermo dei device utilizzati dagli utenti.

Assicurarsi per questo di:

- Non bloccare file javascript, css e file immagini a Googlebot tramite robots.txt;
- Evitare l'inserimento di video o contenuti che non è possibile fruire da device mobile;
- Testare il sito con il tool di Google in modo che non sia inficiata l'usabilità del sito ([Overview](#) | [Lighthouse](#) | [Chrome for Developers](#)).

2.2. Utilizzo dei Breadcrumbs

L'uso di breadcrumbs è fondamentale per facilitare gli utenti e i crawler a comprendere la gerarchia della pagina in cui si trova.

Attraverso i tag microdata di schema.org è possibile facilitare il crawler di Google per capire la struttura corretta del sito e migliorare la visualizzazione nei risultati di ricerca di una pagina, come in questo esempio:

[BreadcrumbList - schema.org](#)
<https://schema.org> > [BreadcrumbList](#) ▾ [Traduci questa pagina](#)
Schema.org Type: **BreadcrumbList** - A **BreadcrumbList** is an **ItemList** consisting of a chain of linked Web pages, typically described using at least their URL and ...

Tutte le voci dei breadcrumbs devono essere linkabili alla relativa pagina, compresa anche l'ultima voce.

2.3. Sitemap e Robots.txt

Sitemap.xml

Per una corretta creazione di una sitemap occorre:

- inserire solo i link canonici di tutte le pagine che si vogliono rendere disponibili ai motori di ricerca;
- non inserire link che rilascino risposte di server differenti dal 200;
- usare il formato XML
- caricare la sitemap nella cartella principale del sito;
- segnalare l'indirizzo della sitemap nel file robots.txt;
- aggiornare dinamicamente gli URL presenti in sitemap.

Robots.txt

L'implementazione corretta del robots.txt prevede:

- caricamento nella folder principale del sito;
- avere solo un file robots.txt;

2.4. URLs

Tutte le URL dovranno essere “*SEO Friendly*” e avere una struttura semplice che possa aiutare l’utente a ricordare la URL di una pagina.

Di seguito alcune norme da seguire:

- Usare i trattini “-” per separare i termini
- Creare URL che rispecchino il contenuto di una pagina
- Evitare la ripetizione di keywords su pagine diverse
- Le url devono tutte essere univoche
- Usare solo caratteri minuscoli

Indicare in ogni pagina il meta tag “canonical” che servirà come unico punto d’accesso alla risorsa richiesta.

2.5. Internalizzazione

I siti dovranno prevedere una struttura delle URLs tale per cui tutte le pagine devono essere inserite sotto due sottocartelle con codice ISO 639-1 per ogni lingua disponibile sul sito ex: “<https://www.autostrade.it/it/la-rete>”

Il redirect su cambio lingua dovrà essere impostato con il codice HTTP 302.

Tutte le pagine dovranno prevedere i tag “*hreflang*” per permettere ai crawler di poter indicizzare una pagina in tutte le versioni che questa presenta, assicurarsi quindi che non ci siano link di ritorno tali per cui se la pagina X rimanda alla pagina Y, a sua volta la pagina Y deve prevedere il tag *hreflang* che rimanda alla pagina X.

2.6. HTTPs

Il sito dovrà essere esposto solamente in HTTPs, usando le più moderne ciphersuites.

2.7. Pagina 404

È necessario prevedere una pagina 404 personalizzata per migliorare la UX globale del sito.

2.8. Performance e pagespeed

I siti dovranno garantire un caricamento completo della pagina entro 3 secondi, ciò comporta che tutto il contenuto “above the fold” deve essere scaricabile in maniera prioritaria evitando tra le altre cose di inserire file javascript all’interno del tag “head” di una pagina. Si consiglia l’utilizzo di

strumenti di sviluppo dedicati per la misurazione di queste performance, ad esempio Google Lighthouse (<https://developers.google.com/web/tools/lighthouse>).

2.9. Headings

Le pagine dovranno, nei limiti del possibile, seguire una struttura degli headings simile a quella utilizzata per gli articoli di giornale per cui:

- Il tag H1 dovrà essere utilizzato solo per il titolo della pagina
- Il tag H2 per i titoli di una sezione o di un articolo, post blog o contenuti simili
- Il tag H3 per eventuali sotto-temi all'interno di una sezione.

Si ricorda inoltre che gli headings dovranno seguire una gerarchia corretta tale per cui h1--> h2--> h3

2.10. Immagini

Le immagini incidono molto sui tempi di caricamento di un sito, si richiede quindi che queste non superino (ove possibile) i 300 Kb, utilizzare solo formati conosciuti quali JPEG, PNG, SVG o WebP. Sono disponibili strumenti online dedicati alla compressione e all'ottimizzazione delle immagini per l'uso su web, a titolo di esempio <https://imagify.io/>.

Le immagini inoltre dovranno anche queste essere nominate in maniera comprensibile sia agli umani che ai bot evitare quindi di utilizzare nomi del tipo: IMG000121.jpeg.

Prevedere su ogni immagine il tag "alt" quale testo alternativo all'immagine al fine di migliorare l'accessibilità e l'indicizzazione dell'immagine.

3. Best Practice Liferay su piattaforma SaaS

Di riportano di seguito le best practice per l'implementazione di Liferay su piattaforma SaaS, basate sulle linee guida generali di Liferay di modo da garantire performance, sicurezza e affidabilità:

- **Sfruttare le capacità di scalabilità del cloud:** Le implementazioni SaaS di Liferay devono essere configurate per sfruttare appieno la scalabilità automatica fornita dai cloud provider. È fondamentale garantire che la piattaforma possa adattarsi dinamicamente ai picchi di traffico e alle variazioni della domanda senza interruzioni;

- **Utilizzare un'architettura modulare con OSGi:** Per garantire la flessibilità e la gestibilità dell'implementazione SaaS, è necessario mantenere un'architettura modulare basata su OSGi, riducendo le dipendenze tra i moduli. Questo permette aggiornamenti e modifiche a singoli componenti senza dover fermare l'intera piattaforma;
- **Centralizzare la gestione delle configurazioni:** In ambienti SaaS, è cruciale implementare una gestione delle configurazioni centralizzata utilizzando strumenti come Liferay Cloud o Kubernetes. Questo garantisce un controllo uniforme delle configurazioni su tutte le istanze e ambienti;
- **Ottimizzare le performance della piattaforma:** È importante adottare tecniche di ottimizzazione delle prestazioni, come l'implementazione di sistemi di caching per contenuti statici e dinamici, la compressione di risorse (CSS, JavaScript) e l'uso di CDN per migliorare i tempi di caricamento delle pagine;
- **Gestire l'isolamento e il multi-tenant:** In un'implementazione SaaS multi-tenant, occorre configurare correttamente Liferay per garantire l'isolamento dei dati e delle configurazioni tra i vari tenant. Questo assicura che ogni cliente o organizzazione mantenga la propria indipendenza e sicurezza;
- **Implementare sicurezza avanzata e protezione dei dati:** Bisogna sfruttare le funzionalità di sicurezza native di Liferay, come il controllo degli accessi basato sui ruoli (RBAC) e la crittografia end-to-end dei dati, sia a riposo che in transito. Queste pratiche sono essenziali per proteggere i dati sensibili e garantire la conformità alle normative come il GDPR;
- **Pianificare backup regolari e disaster recovery:** È fondamentale implementare un sistema di backup automatico e un piano di disaster recovery affidabile. La piattaforma deve essere configurata per eseguire backup periodici di dati e configurazioni, garantendo un ripristino rapido in caso di problemi;
- **Implementare un sistema di monitoraggio e logging centralizzato:** In un ambiente SaaS, è essenziale monitorare costantemente la piattaforma e centralizzare i log tramite strumenti come la Liferay Cloud Console o ELK Stack. Questo permette di rilevare anomalie e risolvere problemi in tempo reale, migliorando l'affidabilità del sistema;
- **Sfruttare l'architettura a microservizi:** L'utilizzo di un'architettura a microservizi consente di sviluppare, distribuire e aggiornare componenti in modo indipendente, riducendo i tempi di inattività e migliorando la gestione del ciclo di vita delle applicazioni in un ambiente SaaS;
- **Effettuare aggiornamenti e patching regolari:** Per garantire la sicurezza e le prestazioni della piattaforma, è importante pianificare aggiornamenti regolari delle versioni di Liferay e

delle patch di sicurezza. Questo assicura che l'implementazione SaaS resti aggiornata e protetta contro le vulnerabilità più recenti.

Per maggiori info sulla piattaforma SaaS Liferay è possibile fare riferimento alla documentazione presente sul sito ufficiale, in particolare:

- Offering e confronto: <https://www.liferay.com/it/offerings#compare-offerings>
- Servizi inclusi: <https://www.liferay.com/it/subscription-services>