



Progetto esecutivo

Relativo all'adesione di Regione Toscana – Settore Sismica alla Accordo quadro per l'affidamento dei “Servizi e prestazioni inerenti la progettazione, sviluppo, configurazione, popolamento ed erogazione in modalità SaaS tramite Sistema Cloud Toscana – SCT, del Front end digitale ovvero dei Siti, Portali, Canali Web e WebApp, di Regione Toscana – Giunta Regionale e degli Enti del territorio regionale” - CIG: 94252656BA

PNRR 1.4.1 - Cittadino Informato



SOMMARIO

Sommario

2

1 32

43

53.1

93.2

103.3

103.4

10



● *Introduzione*

Il presente piano operativo è stato sviluppato per soddisfare i bisogni elencati nel "Piano dei Fabbisogni" del Comune di Firenze. Il piano operativo è stato sviluppato considerando i seguenti fattori:

- Le esigenze espresse
- Le priorità di intervento
- Gli obiettivi da raggiungere



- **Quadro Generale ed Obiettivi del Progetto**

Il "Piano dei Fabbisogni" del Comune di Firenze elenca una serie di bisogni e necessità dell'ente. I bisogni sono suddivisi in 2 macro-categorie:

- **Bisogni strutturali:** riguardano la necessità di interventi di manutenzione e riqualificazione infrastrutturale della "Rete Civica"
- **Bisogni di innovazione:** riguardano la necessità di implementare nuovi servizi e tecnologie per migliorare l'efficienza e l'efficacia della presenza digitale dell'Ente.

Il piano di intervento è stato sviluppato in base alle analisi dei bisogni. Il piano di intervento prevede interventi nelle seguenti aree:

- **Implementazione evolutive:**
 - Implementazione di nuove funzionalità
 - Implementazioni necessarie all'adeguamento della Rete civica ai criteri di conformità dell'avviso PNRR 1.4.1 - cittadino attivo
- **Manutenzione e riqualificazione infrastrutturale della "Rete Civica":**
 - Riqualificazione della codebase e della struttura delle cartelle di Rete Civica



- **Descrizione delle soluzioni progettuali e stima dei costi**

- **REALIZZAZIONE CHILD THEME**

Silfi S.p.A. - Firenze Smart, società in-house del Comune di Firenze, sta già lavorando ad una personalizzazione di un tema generico basato sul framework Bootstrap, nella variante predisposta dall'AgID stessa, il repository è pubblico <https://github.com/municipes/scaenographia>

Nell'ambito del presente progetto verrà creato un sottotema specificatamente per il Comune di Firenze che erediterà le caratteristiche di quanto andrà a sviluppare SILFI e al contempo ne estenderà le funzionalità esclusivamente per le peculiarità del sito della Rete Civica in termini di Content Type, blocchi, regioni.

Riferimenti:

<https://docs.italia.it/italia/designers-italia/design-linee-guida-docs/it/stabile/index.html>

<https://github.com/italia/bootstrap-italia/>

https://www.drupal.org/project/bootstrap_italia

▪

Dismissione del precedente theme

Nel vecchio theme attualmente in utilizzo in Rete Civica sono state inserite delle funzionalità che verranno ripristinate in moduli realizzati ad hoc, in modo da non introdurre regressioni. Si produce qui un elenco a titolo di esempio ma non esaustivo delle funzionalità da replicare:

- Gestione homepage
- Integrazione tra articoli generati sui siti tematici e quelli presenti in Rete civica.
- Integrazione con sito protezionecivile.comune.fi.it, per la visualizzazione in home page delle allerte meteo.

▪

Miscellanea

Per i seguenti punti si rende necessaria la classica iterazione del ciclo di vita del software: Analisi, Progettazione, Programmazione, Debugging, Collaudo, Deployment. Questo ciclo prevederà delle interazioni con il Cliente per la raccolta dei requisiti e l'analisi del dominio applicativo.

- Integrazione col sistema di autenticazione dell'Ente.
- Integrazione col sistema di pagamenti dell'Ente.
- Integrazione col sistema di protocollo dell'Ente.
- Integrazione col sistema di Analytics dell'Ente (Matomo).
- Integrazione col sistema di monitoraggio accessibilità dell'Ente (Siteimprove).
- Modulo di gestione della sezione Selezione del personale.
- Ricerca custom per Articoli e Comunicati stampa.
- Cerimoniale del Sindaco.
- Conferenze stampa e visualizzazione eventi su calendario.



Successivamente alle fasi di analisi verranno formulate le adeguate proposte di programmazione e implementazione.

○ **MIGRAZIONE DEI CONTENUTI VERSO LA NUOVA INFRASTRUTTURA**

Tramite script custom e l'utilizzo del modulo Migrate saranno implementate le procedure di migrazione automatica dei contenuti del vecchio sito al fine di implementare la nuova architettura informativa secondo le linee guida di AGID.

La migrazione dei contenuti sarà agevolata in parte dalla stretta parentela con l'attuale portale, ma sarà comunque possibile importare contenuti da sorgenti anche di natura diversa, con la possibilità ulteriore di mantenere una sincronizzazione con la sorgente ed attuare processi di trasformazione del dato prima che esso sia salvato nel CMS.

In particolare, si svilupperà un processo di migrazione dedicato per ciascuna tipologia di contenuto, adottando le best practice e le API di Migration del framework stesso, e configurando il necessario per rendere automatici i processi di trasferimento dei contenuti. Questo automatismo permetterà di lavorare parallelamente nel vecchio portale e vedere le modifiche trasferite automaticamente sul nuovo portale. Quando lo sviluppo del nuovo portale sarà completato e validato, sarà possibile trasferire il portale senza ulteriore travaso di contenuti.

Nell'ambito del presente progetto verranno analizzate le due strutture dati, sia quella del vecchio portale che quella del nuovo e, in caso di discrepanze, individuate delle policy di migrazione ad hoc.

Per fare questo è previsto l'utilizzo di diversi moduli: MigrationAPI, MigrationTools e/o MigrationPlus.

Di fatto l'ecosistema dei moduli Migrate aggiunge al core di Drupal uno strumento molto potente per processi ETL che mette a disposizione interfacce di programmazione (plugin) per cui la fase di estrazione viene mappata tramite file di configurazione yaml nel plugin source, la fase di trasformazione viene mappata nel plugin process mentre la fase di caricamento (load) viene definita dal plugin destination.

Nella fase di estrazione, un insieme di dati, chiamato riga, viene recuperato dall'origine dati. I dati possono essere migrati da un database, oppure caricati da un file (ad esempio CSV, JSON o XML) o recuperati da un servizio web (ad esempio RSS o REST). Questo caso può verificarsi quando i dati presenti in Drupal necessitano di ulteriori informazioni per poter correlare funzionalità attualmente non presenti. Durante la fase di migrazione ciascuna riga viene inviata alla fase di trasformazione dove viene elaborata secondo necessità o contrassegnata per essere ignorata. Dopo l'elaborazione, la riga trasformata viene passata alla fase di destinazione dove viene caricata (salvata) nel sito Drupal di destinazione.

▪ **Stubs, Map tables, Highwater marks e Rollbacks**

Stubs

Sono dei meccanismi per garantire la consistenza del dato: quando un termine viene migrato, è possibile che il suo termine padre non sia stato ancora migrato. Drupal si preoccupa di inserire un segnaposto che popolerà in un secondo momento quando avrà a disposizione tutte le informazioni.



Map tables

Una volta che una riga migrata è stata salvata e l'ID di destinazione è noto, l'API Migrate salva l'ID di origine, l'ID di destinazione e l'hash della riga in una tabella della mappa. L'ID sorgente e l'hash nella mappa consentono di tenere traccia delle modifiche per le migrazioni continue. La mappa tra l'ID di origine e quello di destinazione consente anche di cercare i valori durante altre migrazioni che vengono eseguite successivamente.

Highwater marks

Consentono a Migrate di tenere traccia delle modifiche in modo da poter migrare solo il contenuto che è stato creato o aggiornato nell'origine poiché la migrazione è stata eseguita in precedenza. Ciò richiede che la sorgente abbia una proprietà che abbia valore incrementale in modo da poter stabilire quale è il valore più alto migrato finora.

Rollbacks

È abbastanza tipico che durante lo sviluppo di una migrazione, la prima versione non fornisca risultati corretti per tutti gli scenari. I rollback consentono di annullare una migrazione, regolarla migrazione e quindi eseguirla di nuovo.

○ ASSEVERAZIONE

La valutazione della qualità del sito istituzionale del Cliente (Rete civica) e della corrispondenza ai criteri di conformità della misura 1.4.1 del PNRR *Esperienza del cittadino nei servizi pubblici digitali* verrà eseguita utilizzando l'applicativo desktop a supporto fornito direttamente da AGID <https://github.com/italia/pa-website-validator> o in alternativa la sua controparte stand alone: <https://agid.webapplication.me>

Si specifica che PA Website Validator è uno strumento che integra la libreria [Lighthouse](#) fornita da Google a corredo di Chrome: <https://developer.chrome.com/docs/lighthouse/overview/>

Si fa inoltre presente che Lighthouse è uno strumento standard spesso utilizzato per le verifiche che precedono la pubblicazione o messa in produzione di un progetto.

Questo perché le PM (*Performance Metrics*) e i CWV (*Core Web Vitals*) oltre che essere la cartina di tornasole della user experience, sono un importante fattore di ranking, ovvero incidono fortemente sul posizionamento di un sito sui motori di ricerca più importanti.

Di seguito riportiamo le indicazioni che seguiamo e gli accorgimenti che adottiamo per migliorare questi fattori.

Contiamo molto sul fatto che l'attività di ASSEVERAZIONE venga facilitata dalle *best practice* sopra descritte.

▪ **Metriche Performance e Core Web Vitals**

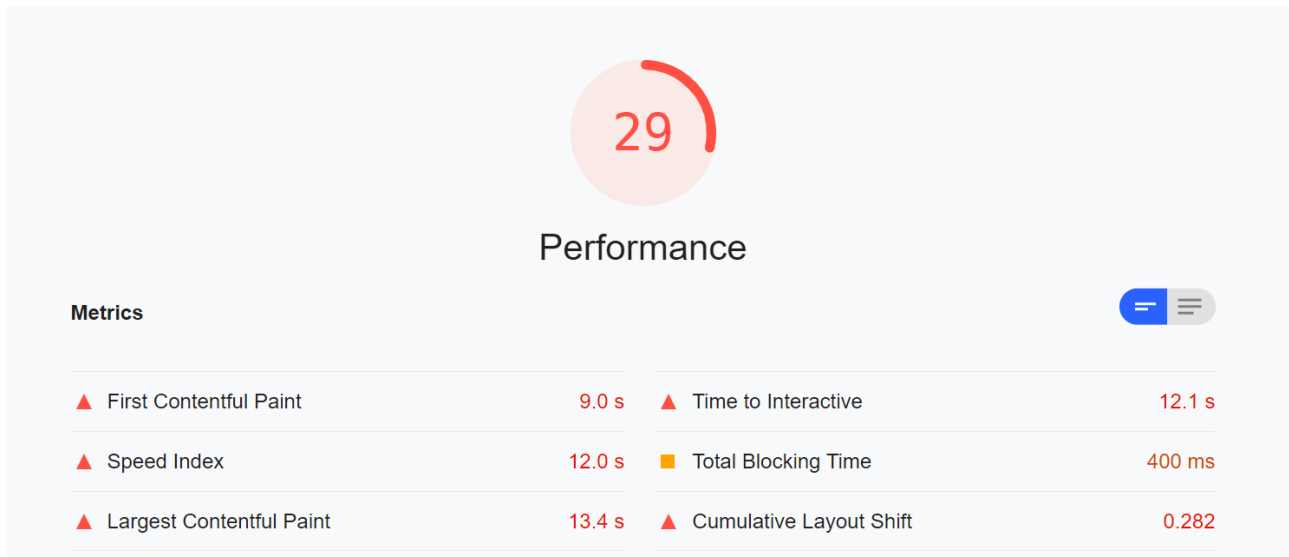
È necessario rispettare i Core Web Vitals, set di specifici indicatori considerati fattori di ranking, strutturare quindi il nuovo sito con il fine di ottimizzare le performance online, dal punto di vista della user experience, velocità, tempo di risposta e stabilità del layout.

Nel dettaglio i tre fattori che consideriamo solitamente sono:



- LCP: Largest Contentful Paint - Velocità di caricamento (loading) - indica il tempo di rendering dell'elemento più grande, per misurare il tempo di attesa percepito dagli utenti.
- FID: First Input Delay - Interattività della pagina (interactivity)
- CLS: Cumulative Layout Shift - Stabilità visiva del caricamento (visual stability)

Esempio:



LCP: una buona esperienza utente dovrebbe rientrare nei 2,5 secondi (meno di 2,5 secondi per almeno 75 delle sessioni sulla pagina). Sopra siamo su una pagina che non ha una buona attività di ottimizzazione soprattutto sugli elementi che impegnano gran parte delle risorse come immagini, video, eventuali background. E' importante ottimizzare tale metrica del caricamento, per una migliore esperienza utenti possibile.

E' suggerito utilizzare Link Prefetching, consente di precaricare risorse, salvarle in cache e renderle disponibili quando l'utente le richiede.

FID: superati i 300 millisecondi andiamo verso una scarsa ottimizzazione della navigabilità, può portare all'abbandono della pagina e/o a un mancato rilascio del contatto.

CLS: un risultato superiore a 0.25 è da migliorare, esperienza negativa con conseguente possibile abbandono della pagina.

I principali fattori che influenzano negativamente il tempo di generazione della pagina HTML (velocità di caricamento delle pagine) sono la performance del server e la densità del codice HTML della pagina. È consigliabile "pulire" il codice HTML della pagina.

Le dimensioni HTML di una pagina sono le dimensioni di tutto il codice HTML in essa contenuto. Dimensioni di pagina troppo grandi (ovvero, superiori a 2 MB) causano un tempo di caricamento più lento con una conseguente esperienza utente negativa e un ranking più basso tra i risultati dei motori di ricerca.

→ Minimizzare il codice, implementare la cache e rimuovere il codice non utilizzato.



Si può prendere in considerazione anche la suddivisione del codice in blocchi che vengono eseguiti solo quando necessari, invece di richiedere al browser di caricarli tutti prima che la pagina diventi interattiva.

→ Lavorare sulle immagini, riducendo peso e dimensioni e utilizzando formati più moderni come ad esempio il WebP. Aggiungere l'attributo `loading=lazy` alle immagini che non sono nel viewport iniziale, per segnalare al browser che è possibile ritardare il caricamento di queste immagini perché hanno meno probabilità di essere viste subito.

→ Utilizzare la cache del browser per ridurre i tempi di caricamento delle pagine e aumentare le prestazioni. Migliore user experience e diminuzione del tasso di uscita dalle pagine. → Diminuire il numero di richieste effettuate dalla pagina eliminando file JavaScript e fogli di stile CSS quando non servono.

→ Eliminare risorse di blocco della visualizzazione in quanto rallentano i tempi di caricamento percepiti delle pagine del sito Wp. Ritardare quindi il caricamento delle risorse che non sono immediatamente necessarie.

I principali motori di ricerca premiano i siti che offrono una migliore esperienza utente, è bene quindi utilizzare il protocollo HTTP/2, riduce i tempi di caricamento della pagina migliorando l'efficienza delle comunicazioni tra server e browser.

Altri accorgimenti che mettiamo in campo solitamente:

- parallelizzazione del caricamento dei css tramite `preload/prefetch`
- parallelizzazione del caricamento del javascript tramite `async` o `defer`
- caricare i font con il parametro `<link rel=preload>` .
- Riduzione della dimensione delle immagini più grandi

○ **PREDISPOSIZIONE NUOVA RETE CIVICA**

L'obiettivo principale di questo intervento è quello di rendere la Rete civica pienamente conforme ai criteri dell'avviso PNRR 1.4.1 - cittadino attivo, valutando la possibilità di evolvere verso un'architettura a *microservizi* e verso tecnologie *cloud native*. Il dettaglio dell'implementazione sarà concordato a seguito dell'analisi del contesto e delle esigenze del Comune.

3.1 L'approccio progettuale

Il progetto sarà gestito attraverso l'applicazione dei principi AGILE – SCRUM.

Questa metodologia prevede il coinvolgimento degli utenti che utilizzeranno il software fin dalla prima fase di sviluppo e l'impostazione del lavoro secondo cicli di rilascio definiti, che garantiscono al Cliente la validazione di porzioni di applicazione realmente aderenti alle esigenze degli utenti e immediatamente testabili dagli stessi.

Al fine di adottare una metodologia di lavoro DevOps, il fornitore si interfacerà coi sistemi del Comune di Firenze attraverso un'istanza GitLab self-hosted, principalmente per i seguenti scopi:

- rilasci e versionamento del codice
- definizione del piano di dettaglio



- conduzione
- CI/CD.

3.2 Gruppo di Lavoro e CV

Il gruppo di lavoro messo a disposizione del Comune di Firenze per l'esecuzione del piano operativo e i relativi CV verranno condivisi con l'amministrazione in fase operativa, ad ogni modo i profili saranno in linea con quelli previsti dalla convenzione.

Le attività saranno svolte dalla azienda Engineering ingegneria informatica che potrà avvalersi, come previsto dalla Convenzione del Subappalto.

3.3 Dettaglio Costi

Tutti i valori sono da considerarsi Iva esclusa

Figura Professionale	Prezzo unitario gg/p	N. GG/UOMO	Totale
1 – Capo Progetto (Client + Project Manager)	450	40	18.000,00 €
2- Amministratore di sistema senior	350	40	14.000,00 €
3 – Analista funzionale Senior	350	35	12.250,00 €
4- Analista Programmatore Senior	330	80	26.400,00 €
5- Analista Programmatore Junior	250	0	0.000,00 €
6- Web Graphic Designer	270	60	16.200,00 €
7- Esperto User Interface desing, usability testing, graphics design	340	60	20.400,00 €
8- Copy Writer	250	3	750,00 €
9- Esperto Accessibilità (UX/UI)	350	100	35.000,00 €
10 – Esperto Social Media	350	0	0.000,00 €
11- Esperto ESPERTO PRIVACY E TRATTAMENTO DATI	350	20	7.000,00 €
Totale			150.000,00 €

3.4 Piano di lavoro

Di seguito riportiamo il piano di lavoro che riguarda la pianificazione della progettazione esecutiva che è la prima fase del progetto, nel rispetto di quanto indicato nel piano dei fabbisogni.

Tutte le successive attività verranno stimate e pianificate sulla base del cronoprogramma definito nella progettazione esecutiva; nel corso del contratto la pianificazione potrà subire variazioni sulla base di mutate esigenze dell'amministrazione.

Cronoprogramma

Attività	15/01/24	15/01/24	31/01/24	31/01/24	31/01/24	28/02/24	15/03/24	30/04/24	30/06/24	30/11/24
Prima versione stabile del child theme in ambiente demo										



Conclusione definizione regole di migrazione		■								
Conclusione esecuzione della prima migrazione in <i>demo</i>			■							
Primo ciclo verifiche con app fornita dal DTD e manuali				■						
Valutazione accessibilità e dichiarazione di conformità					■					
Test finali prima della messa in produzione						■				
Trasferimento in ambiente di produzione/Go live							■			
Prima asseverazione								■		
Correzioni/adeguamenti									■	
Termine ultimo asseverazione finale (con esito positivo)										■