

DATABASE SISMOGRAMMI STORICI VESUVIANI

a cura di G. Ferricksi, L. Di Rita, F. Di Giuseppe e G. Ricciardi

Home Database



In questa pagina sono raccolte le scansioni digitali di numerosi sismogrammi storici registrati al Vesuvio nel periodo compreso fra il 1915 ed il grosso interesse scientifico, oltre che storico, essendo le uniche registrazioni sismiche disponibili del Vesuvio in fase eruttiva. In particolare, si tratta di sismogrammi registrati in condizioni di campo aperto, caratterizzate da terreno vulcanico, ovvero la vibrazione continua del suolo causata dal movimento sismico, il segnale causato da piccole esplosioni che si verificano a distanza di pochi minuti l'una dall'altra e coinvolgono l'intero

Inoltre alcuni dei sismogrammi presenti nel database sono stati registrati proprio nel corso dell'eruzione del 1944, fornendo informazioni preziose

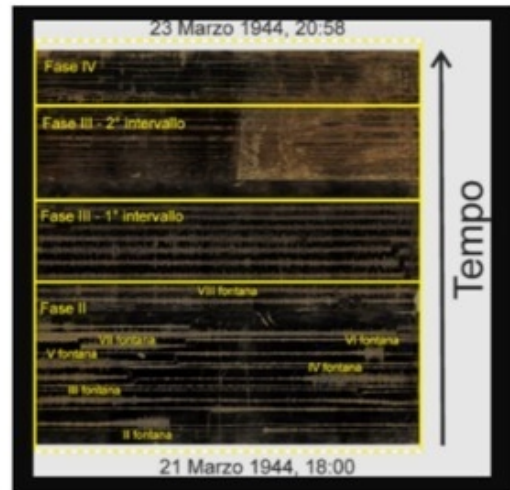
Questi sismogrammi hanno consentito, ad esempio, lo studio delle fasi parossistiche dell'eruzione del Vesuvio del 1944, fornendo informazioni preziose sul comportamento del Vesuvio durante le fasi eruttive più violente (Pappalardo et al., 2014).

Per maggiori dettagli: "Riproduzione digitale di sismogrammi storici del Vesuvio. Prime registrazioni sismiche su un vulcano" (Rapporti tecnici INGV 2015, numero 294).

Ortosismografo Alfani

DATABASE SISMOGRAMMI STORICI VESUVIANI

a cura di G. Ferricksi, L. Di Rita, F. Di Giuseppe e G. Ricciardi



Questi sismogrammi hanno consentito, ad esempio, lo studio delle fasi parossistiche dell'eruzione del Vesuvio del 1944, fornendo informazioni preziose sul comportamento del Vesuvio durante le fasi eruttive più violente (Pappalardo et al., 2014).

Per maggiori dettagli: "Riproduzione digitale di sismogrammi storici del Vesuvio: Prime registrazioni sismiche su un vulcano" (Rapporti tecnici INGV 2015, numero 294).

RAPPORTI TECNICI INGV

Database Sismogrammi Storici Vesuviani: reingegnerizzazione del portale web con RWD (Responsive Web Design)



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

439

Esempio di registrazione dell'attività sismica al Vesuvio

Tali scansioni si distinguono perché evidenziano che si sono registrate e ricorrono ad un periodo ben preciso.

In questa pagina sono raccolte le scansioni digitali di numerosi sismogrammi storici, registrati al Vesuvio nel periodo compreso fra il 1915 ed il 1944. Tali sismogrammi sono di grosso interesse scientifico, oltre che storico, essendo le uniche registrazioni sismiche disponibili del

Direttore Responsabile

Valeria DE PAOLA

Editorial Board

Luigi CUCCI - Editor in Chief (luigi.cucci@ingv.it)
Raffaele AZZARO (raffaele.azzaro@ingv.it)
Christian BIGNAMI (christian.bignami@ingv.it)
Mario CASTELLANO (mario.castellano@ingv.it)
Viviana CASTELLI (viviana.castelli@ingv.it)
Rosa Anna CORSARO (rosanna.corsaro@ingv.it)
Domenico DI MAURO (domenico.dimauro@ingv.it)
Mauro DI VITO (mauro.divito@ingv.it)
Marcello LIOTTA (marcello.liotta@ingv.it)
Mario MATTIA (mario.mattia@ingv.it)
Milena MORETTI (milena.moretti@ingv.it)
Nicola PAGLIUCA (nicola.pagliuca@ingv.it)
Umberto SCIACCA (umberto.sciacca@ingv.it)
Alessandro SETTIMI (alessandro.settimi1@istruzione.it)
Andrea TERTULLIANI (andrea.tertulliani@ingv.it)

Segreteria di Redazione

Francesca DI STEFANO - Coordinatore
Rossella CELI
Robert MIGLIAZZA
Barbara ANGIONI
Massimiliano CASCONI
Patrizia PANTANI
Tel. +39 06 51860068
redazione@ingv.it

REGISTRAZIONE AL TRIBUNALE DI ROMA N.174 | 2014, 23 LUGLIO

© 2014 INGV Istituto Nazionale
di Geofisica e Vulcanologia
Rappresentante legale: Carlo DOGLIONI
Sede: Via di Vigna Murata, 605 | Roma



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

RAPPORTI TECNICI INGV

Database Sismogrammi Storici Vesuviani:
reingegnerizzazione del portale web
con RWD (Responsive Web Design)

*Vesuvian Historical Seismograms Database,
web portal reengineering using RWD
(Responsive Web Design)*

Francesca Cirillo¹, Rosario Peluso¹, Oussama Ferchichi²

¹INGV | Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Napoli - Osservatorio Vesuviano

²Università degli Studi di Salerno

Accettato 7 settembre 2021 | Accepted 7 September 2021

Come citare | How to cite Cirillo F., Peluso R. e Ferchichi O., (2022). Database Sismogrammi Storici Vesuviani: reingegnerizzazione del portale web con RWD (Responsive Web Design). Rapp. Tec. INGV, 439: 1-36, <https://doi.org/10.13127/rpt/439>

In copertina Portale Sismogrammi Storici responsive: viste desktop e mobile | Cover Historical seismograms responsive portal: desktop and mobile view

439

INDICE

Riassunto	7
<i>Abstract</i>	7
Introduzione	8
1. Portale Database Sismogrammi Storici Vesuviani	8
1.1 I Sismogrammi Storici	8
1.2 Problematiche e soluzioni	9
2. Portale Web <i>Responsive</i>	10
2.1 Il vecchio portale	10
2.2 Reingegnerizzazione interfaccia <i>desktop</i> e <i>mobile</i>	11
2.2.1 Pagina <i>Home</i>	12
2.2.1.1 Barra di Navigazione	12
2.2.1.2 <i>Card</i>	13
2.2.2 Pagina <i>Database</i>	14
2.2.2.1 Barra di Ricerca	15
2.2.2.2 Tabella	15
2.2.2.2.1 Ordinamento della tabella	16
2.2.2.3 Barra di Paginazione	16
2.3 Pagina <i>Preview</i>	17
3. Framework <i>client-side</i> di sviluppo: <i>Bootstrap</i>	17
3.1 Alcuni concetti generali	17
3.2 <i>Framework client-side</i>	18
3.2.1 <i>Bootstrap</i>	19
3.2.1.1 <i>Layout</i>	19
3.2.1.2 Componenti	20
3.2.1.3 Altri Strumenti	22
4. Implementazione	22
4.1 File e organizzazione in cartelle	23
4.2 Barra di Navigazione	23
4.3 <i>Card</i>	24
4.4 Barra di ricerca	24
4.4.1 <i>Alert</i>	25
4.5 Tabella	25
4.5.1 Generazione dati in formato JSON	26
4.5.2 Ordinamento della tabella	27
4.6 Barra di Paginazione	28
4.6.1 Selezione pagina	29
4.6.2 Gestione della parte <i>responsive</i>	29
4.7 <i>Preview</i>	30
5. Conclusioni	31
Bibliografia	32
Sitografia	32

Riassunto

Nell'ambito del Tirocinio svolto in collaborazione con l'Università degli Studi di Salerno, Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata, è stato reingegnerizzato il vecchio portale "Database Sismogrammi Storici Vesuviani" al fine di rendere accessibile e diffondere, senza alcun limite, il patrimonio storico e scientifico dei Sismogrammi Storici Vesuviani.

Tale portale, una volta all'avanguardia, era oramai caratterizzato da tecnologie/metodologie datate e soprattutto non accessibili dai nuovi dispositivi mobili. In considerazione del fatto che oggi 9 italiani su 10 usano lo smartphone per navigare su internet, si è ritenuto necessario, oltre che doveroso verso la comunità scientifica e non, di procedere con la reingegnerizzazione del portale attraverso un approccio di tipo Responsive Web Design. L'applicazione di tale approccio ha consentito di ottimizzare la presentazione del portale in base alle dimensioni dello specifico dispositivo utilizzato per la fruizione dei contenuti (desktop, tablet, smartphone).

Sono state inoltre implementate delle funzionalità che consentono una più agile e mirata ricerca del sismogramma desiderato tra i circa 300 sismogrammi esistenti: in precedenza si poteva effettuare unicamente una ricerca di tipo sequenziale.

Il nuovo portale è stato implementato attraverso l'utilizzo del framework di sviluppo open source *Bootstrap* che grazie ai componenti che questo mette a disposizione dello sviluppatore, ha consentito di realizzare in maniera più veloce ed ottimizzata il nuovo portale Responsive di tipo Mobile-First.

Tale portale è disponibile al seguente indirizzo web: <http://portale2.ov.ingv.it/carte/>

Abstract

As part of the internship carried out in collaboration with the University of Salerno, Department of Information and Electrical Engineering and Applied Mathematics, the old portal "Vesuvian Historical Seismograms Database" was reengineered in order to make accessible and disseminate without any limit, the historical but also scientific heritage of the Vesuvian Historical Seismograms.

This portal, once at the forefront, is now characterized by outdated technologies / methodologies and it is not accessible by new mobile devices. At present 9 out of 10 Italians use smartphones to surf the internet, it was considered necessary as well as a duty to the scientific and non-scientific community, to proceed with the re-engineering of the portal through a Web Design Responsive approach. The application of this approach made it possible to optimize the portal presentation on all access device (desktop, tablet, smartphone).

In addition, a functionality has been implemented, it allow a more agile and targeted search for the desired seismogram among the approximately 300 existing seismograms: previously, only a sequential search could be performed.

The new portal was implemented using the open source development framework Bootstrap. Thanks to the components that it offers to the developer, it made possible to create the new Mobile-First Responsive portal in a faster and more optimized way.

This portal is available at the following web address: <http://portale2.ov.ingv.it/carte/>

Keywords Responsive Web Design; Bootstrap; Sismogrammi Storici Vesuviani | Vesuvian Historical Seismograms

Introduzione

L'Osservatorio Vesuviano è il più antico osservatorio vulcanologico al mondo: per tale motivo conserva un'instimabile ricchezza in termini di strumenti storici utilizzati per le registrazioni sismiche nonché di sismogrammi storici (intesi come tracciati dell'evoluzione temporale delle onde sismiche).

Circa 300 sismogrammi, alcuni dei quali risalenti ai primi del 900, sono stati digitalizzati e memorizzati nel "Database Sismogrammi Storici Vesuviani", raggiungibile attraverso un vecchio portale caratterizzato da tecnologie oramai in disuso e non accessibile dai dispositivi mobili.

Al fine di condividere questo importante patrimonio senza alcuna restrizione e per migliorare le funzionalità di accesso e di filtraggio dei dati in esso contenuti (i sismogrammi), è stato realizzato un nuovo portale di tipo *responsive*, cioè caratterizzato da un'interfaccia grafica che si adatta a tutti i dispositivi: *PC desktop*, *smartphone* e *tablet*.

Il seguente documento è così strutturato:

- una prima sezione che presenta una *panoramica del vecchio portale* in cui si evidenziano svantaggi e limitazioni (molte delle quali connesse alla visualizzazione sui dispositivi mobili). Ad essa fa seguito una presentazione della soluzione individuata per la realizzazione del nuovo portale basata su un approccio di tipo *Responsive Web Design*;
- una sezione per la presentazione dell'*interfaccia* e di tutte le componenti che la caratterizzano (sia in ambito *desktop* che *mobile*), delle varie funzionalità e delle modalità attraverso cui è possibile usufruirne;
- una sezione per la presentazione delle tecnologie utilizzate e del *framework* selezionato;
- una sezione in cui vengono mostrati tutti i "dettagli implementativi".

Si è preferito non unire aspetti legati all'interfaccia con quelli di tipo più implementativo, in maniera tale da consentire una lettura più lineare (non appesantita da aspetti legati allo sviluppo) ma soprattutto in modo da differenziare le varie viste, la prima di interesse generale (interfaccia, funzionalità e navigazione) la seconda di interesse per gli sviluppatori o per chi desidera approfondire anche gli aspetti di carattere implementativo, che nonostante siano stati presentati nella forma più semplice possibile, richiedono almeno una conoscenza di base di argomenti quali *HTML*¹, *CSS*², costrutti basilari della programmazione (*for*, *if*, ecc.) e concetto di Classe ed Ereditarietà.

1. Portale Database Sismogrammi Storici Vesuviani

1.1 I Sismogrammi Storici

L'Osservatorio Vesuviano è stato fondato nel 1841 dal re delle due Sicilie Ferdinando II di Borbone, ed è il più antico osservatorio vulcanologico del mondo. La sua sede originaria è ubicata sul Vesuvio, sul Colle del Salvatore a 608 metri di quota, luogo reputato al tempo particolarmente adatto, in quanto sufficientemente distante dal cratere da non essere raggiunto dai lapilli e dai proietti di grosse dimensioni, e abbastanza elevato da non essere interessato dalle colate di lava. Grazie alle sue antiche strumentazioni e al continuo lavoro di monitoraggio, sono arrivati fino a noi importanti documenti, non solo di interesse storico ma anche scientifico. Tra di essi ci sono,

¹ *HTML (HyperText Markup Language)* è il linguaggio utilizzato per creare pagine web e altri tipi di documenti visualizzabili in un browser.

² *CSS* sta per *Cascading Style Sheets*, in italiano "fogli di stile". È un linguaggio con cui si gestisce il design e la presentazione delle pagine web (dal punto di vista estetico) e lavora in combinazione con l'*HTML*, o *HyperText Markup Language* (linguaggio a marcatori per ipertesti, il quale invece gestisce i contenuti delle pagine).

ad esempio, i sismogrammi prodotti con l'ortosismografo Alfani, costituito da una sbarra d'acciaio a cui era collegata una massa oscillante che pesava 200 Kg. I sismogrammi³ venivano registrati su carte affumicate, posizionate su un tamburo rotante ed incisi da un pennino di alluminio mosso dalle oscillazioni della massa.

Al fine di condividere una parte importante di questo patrimonio culturale e scientifico, alcuni ricercatori dell'Osservatorio Vesuviano hanno provveduto alla digitalizzazione di oltre 300 sismogrammi storici vesuviani registrati nel periodo compreso tra il 1915 ed il 1944 [Fiore S. et al., 2015]. In considerazione del periodo di riferimento, tra essi è possibile trovare quella che con molta probabilità è la più antica registrazione sismica pervenuta fino a noi su un vulcano (17 settembre 1915) oltre alle registrazioni sismiche del Vesuvio in fase eruttiva. Questi ultimi sismogrammi hanno consentito anche lo studio delle fasi parossistiche dell'eruzione del Vesuvio del 1944, fornendo informazioni preziose sul comportamento del Vesuvio durante le fasi eruttive più violente.

Tali scansioni digitali sono state raccolte nel "Database Sismogrammi Storici Vesuviani" e sono state pubblicate attraverso un portale accessibile a tutti (Figura 1).

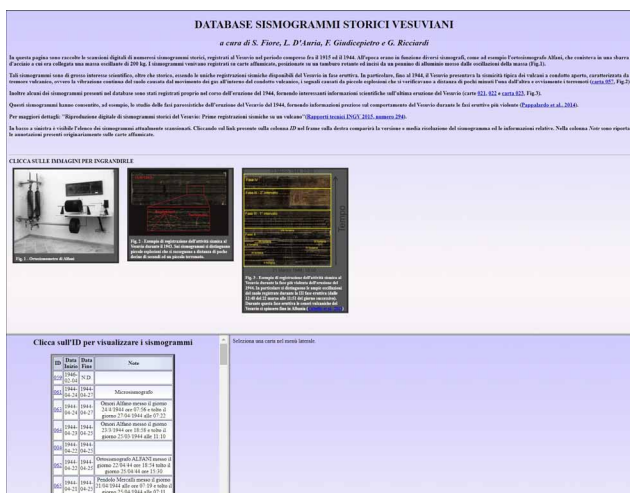


Figura 1 Vecchio Portale.

Figure 1 Old Portal.

1.2 Problematiche e soluzioni

Il vecchio portale citato nella precedente sezione, un tempo innovativo e fruibile da tutti, attualmente presentava una serie di problematiche relative all'utilizzo di vecchie tecnologie non più al passo con i tempi.

Il web, infatti, ha subito negli ultimi anni evoluzioni molto profonde, molte delle quali connesse alla nascita dei dispositivi mobili. Oramai l'utilizzo di smartphone e tablet è sempre più radicato: basti pensare che ci sono oltre 80 milioni di smartphone per una popolazione di 60 milioni⁴ (con un rapporto, quindi di circa 1,3 telefoni mobili per persona, in media). Risulta inoltre che il 94% delle persone possiede uno *smartphone* mentre solo il 77% possiede un *computer* (*laptop* o *desktop*), infine, secondo uno studio ISTAT⁵ 9 italiani su 10 usano lo *smartphone* per navigare su *internet*.

Premessi questi numeri, è oggi necessario avere dei portali/siti che rispondano in maniera

³ Un sismogramma è un grafico risultante dalle registrazioni fatte da un sismografo, che può rappresentare lo spostamento, la velocità o l'accelerazione del suolo in funzione del tempo.

⁴ <https://wearesocial.com/digital-2020> (Report annuale 2020)

⁵ <https://www.istat.it/> 18/12/2019

adeguata e ottimizzata quando si accede da un dispositivo mobile. Questi ultimi hanno infatti dei *display* più piccoli dei *computer* e si utilizzano tramite *touch screen*. I siti web che non si possano scalare o che contengano elementi grafici di dimensioni elevate, non vengono visualizzati in maniera proporzionata e adeguata sui dispositivi mobili. Come conseguenza logica, questi siti vengono anche esclusi dai risultati di ricerca o vengono mostrati come inadatti.

Poiché il vecchio portale, presentato nella precedente sezione, non era fruibile da dispositivi mobili, è nata l'esigenza di reingegnerizzarlo e di realizzarne uno di tipo *responsive*. Un sito viene definito *responsive* quando è basato su un template grafico che adotta i principi del *responsive design*, una tecnica di progettazione grazie alla quale ogni elemento della pagina si adatta automaticamente per essere visualizzato nel modo migliore su un qualsiasi tipo di dispositivo (*computer, tablet, smartphone, ecc.*).

Attraverso tale reingegnerizzazione si è ottenuto un portale sicuramente al passo con i tempi, più fruibile e caratterizzato da una leggibilità garantita e ottimizzata dei testi e dei contenuti da qualsiasi dispositivo ci si acceda.

2. Portale Web *responsive*

2.1 Il vecchio portale

Il vecchio portale⁶ era costituito da una sola pagina *web* con le tre sezioni di seguito presentate:

- *sezione Introduzione*: caratterizzata da una descrizione dello scopo del portale e dei suoi contenuti, da alcuni esempi di registrazioni sismiche ed infine dalle istruzioni per l'utilizzo delle altre due sezioni (Figura 2).

Figura 2 Sezione Introduzione.
Figure 2 Introduction Section.



- *Sezione Tabella*: sotto la sezione introduzione, separata da una linea orizzontale, e collocata sul lato sinistro si trovava la seconda sezione. Essa presentava una tabella con i dati delle carte associate ai sismogrammi. Tale tabella era caratterizzata da 4 colonne: ID, Data Inizio, Data Fine e Note (Figura 3 a sinistra). Cliccando sull'ID il corrispondente sismogramma veniva presentato nella terza sezione, riportata alla destra di questa e separata da una linea verticale;
- *sezione Preview*: in questa sezione veniva mostrata l'immagine della carta (selezionata nella precedente sezione) e una tabella che riportava la Data di inizio e di Fine e le eventuali Note ad essa associate (Figura 3 a destra).

I problemi relativi a tale interfaccia sono riassunti di seguito:

- poco funzionale e intuitiva: richiedeva istruzioni per il suo utilizzo;
- utilizzava i frame: oramai deprecati;

⁶ <http://sismolab.ov.ingv.it/carte/> data accesso 13.04.2021

- non *responsive*: inutilizzabile sui dispositivi con schermi piccoli;
- non consentiva di effettuare ricerche relative ai sismogrammi nella tabella. La ricerca era solo sequenziale e quindi risultava scomoda da effettuare in considerazione del numero di elementi presenti (300).



Figura 3 Sezione Tabella e Sezione *Preview*.
Figure 3 Table Section and Preview Section.

2.2 Reingegnerizzazione interfaccia *desktop* e *mobile*

Come descritto nella precedente sezione, il vecchio portale era caratterizzato da una singola pagina divisa in 3 frame. Il nuovo portale invece è caratterizzato da una semplice struttura ad albero rappresentata nella seguente figura (Figura 4).

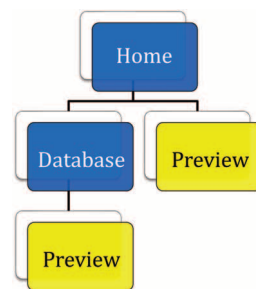


Figura 4 Struttura del nuovo portale.
Figure 4 New Portal Structure.

Nello specifico il portale è costituito da due pagine *web* accessibili attraverso una barra di navigazione:

- la prima pagina *web* **Home** contiene: la barra di navigazione, il testo di presentazione del portale e tre *card* (una fornisce informazioni specifiche sui dispositivi utilizzati per le registrazioni dei sismogrammi e due ne riportano concreti esempi) (Sezione 2.2.1);
- la seconda pagina *web* **Database** contiene: la barra di navigazione, la barra di ricerca per la tabella e la tabella stessa (Sezione 2.2.2).

A partire da entrambe le pagine è possibile accedere ai dettagli di un sismogramma attraverso una **finestra a comparsa** che chiameremo **Preview** (Sezione 2.3). Tale finestra potrà essere aperta in più modi, attraverso un clic:

- sui link testuali presenti nella *Home*;
- sulle immagini dei sismogrammi presenti nelle *card* della *Home*;
- sulle righe della tabella nella pagina *Database*.

Una volta aperta, la pagina *Preview* andrà a coprire quella di partenza. La sua chiusura sarà possibile cliccando sul pulsante a forma di "X" o sull'*overlay* scuro che copre la pagina sottostante (Sezione 2.3).

Tutti i componenti del portale sono stati resi *responsive*, attraverso soluzioni differenti, in maniera tale da offrire una interfaccia semplice ed elegante senza perdita di informazioni anche su dispositivi di dimensioni ridotte.

Le seguenti sotto-sezioni presentano brevemente le modalità di navigazione all'interno delle pagine e le soluzioni adottate per rendere le singole componenti del portale *responsive*. La trattazione è stata differenziata per coloro che sono interessati unicamente agli aspetti connessi alla navigazione e alle funzionalità offerte, e per quelli che intendono approfondire anche gli aspetti relativi allo sviluppo, questo anche al fine di non appesantire il testo. Per questo motivo tutti gli argomenti connessi all'implementazione, sono stati trattati in una sezione a parte (Sezione 4). Si fa notare che nelle Sezioni 2 e 4 sono presenti delle sotto-sezioni denominate allo stesso modo, ciò al fine di evidenziare che queste trattano aspetti differenti (interfaccia/implementazione) legati alla stessa pagina/funzionalità/elemento grafico.

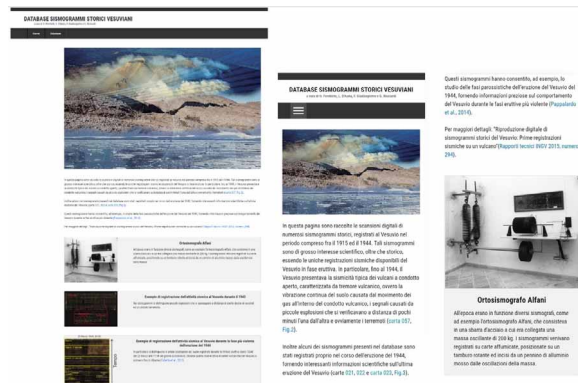
2.2.1 Pagina Home

Come già brevemente accennato, la **Home** (Figura 5) disponibile all'indirizzo <http://portale2.ov.ingv.it/carte/>, è caratterizzata da una **barra di navigazione** in alto, una immagine decorativa seguita dal **testo** introduttivo ed infine tre **card** con rispettive immagini.

La prima **Card** illustra con quali dispositivi avvenivano le registrazioni dei sismogrammi, mentre le altre **Card** presentano degli specifici esempi.

Di seguito viene riportata l'immagine della pagina **Home** sia in caso di navigazione *desktop* (Figura 5 a sinistra) che *mobile* (Figura 5 a destra).

Figura 5 Home page desktop e mobile.
Figure 5 Desktop and mobile home page.



Per il **testo** di introduzione al portale, si è fatto in modo che occupi sempre tutta la larghezza del dispositivo finché non raggiunge una determinata larghezza che ferma la sua espansione.

2.2.1.1 Barra di Navigazione

Per la gestione della barra di navigazione in caso di dispositivi *desktop* è stata adottata una soluzione tradizionale che prevede il titolo del portale ed un elenco orizzontale di link (**Home** e **Database**) attraverso cui muoversi tra le pagine (Figura 6).



Figura 6 Barra di navigazione desktop.
Figure 6 Desktop navigation bar.

Per i dispositivi mobili l'elenco di link viene sostituito da un pulsante (comunemente chiamato *Hamburger Menù*⁷) (Figura 7) che permette l'apertura di una *sidenav*⁸ (Figura 8).



Figura 7 Barra di navigazione *mobile*.

Figure 7 Mobile navigation bar.

Quando si clicca sul pulsante, la *sidenav* presenta l'elenco di link in forma verticale e un pulsante a forma di "X" attraverso cui, con un semplice clic sarà possibile chiuderla (Figura 8).



Figura 8 *Sidenav* per la navigazione da dispositivi mobili.

Figure 8 *Sidenav* for browsing from mobile devices.

Per i dettagli implementativi fare riferimento alla Sezione 4.2.

2.2.1.2 Card

La soluzione adottata per la visualizzazione delle *card* prevede un semplice cambio di disposizione, cioè una *card* orizzontale per i dispositivi *desktop* (Figura 9) e una verticale per i dispositivi *mobile* (Figura 10).

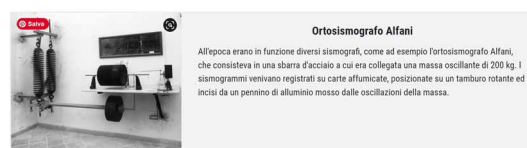


Figura 9 *Card* per i dispositivi *desktop*.

Figure 9 *Card* for desktop devices.

⁷ L'*Hamburger Menù* è un tipico bottone nato dall'esigenza di racchiudere in un'unica icona, che occupasse il minor spazio possibile, tutto il menù delle pagine di un sito o di un'app.

⁸ La *sidenav* è una "barra laterale di navigazione" ed è un termine che si usa per definire una sezione grafica di un sito internet, normalmente posizionata a destra o a sinistra della pagina principale e che racchiude i bottoni di navigazione.

Figura 10 Card per i dispositivi mobili.
Figure 10 Card for mobile device.



Per i dettagli implementativi fare riferimento alla Sezione 4.3.

2.2.2 Pagina Database

La seconda pagina, denominata **Database**, presenta nell'ordine: una **barra di navigazione**, una **barra di ricerca** per i sismogrammi, una **tabella** e una **barra di paginazione** (Figura 11 e Figura 12). L'utente può navigare all'interno della pagina interagendo in vari modi:

1. con la **barra di navigazione**. Descritta nella sezione 2.2.1.1;
2. con la **barra di ricerca**: l'utente può digitare il testo da usare per filtrare le righe della tabella, inviare la richiesta attraverso il pulsante a forma di lente di ingrandimento e annullare la ricerca/filtro attraverso il pulsante a forma di "X" (Sezione 2.2.2.1);
3. con la **tabella**: l'utente attraverso un clic sull'intestazione delle colonne può cambiare l'ordine delle righe della tabella (Sezione 2.2.2.2);
4. con la **barra di paginazione**: l'utente attraverso un clic sui pulsanti della barra di paginazione può navigare tra le varie pagine della tabella (Sezione 2.2.2.3);
5. con la **preview**: l'utente attraverso un clic su una delle righe della tabella può aprire una pagina in cui può visualizzare il sismogramma selezionato (Sezione 2.3).

Di seguito le immagini della pagina **Database** sia in caso di navigazione *desktop* (Figura 11) che *mobile* (Figura 12).

Figura 11 Database page per i dispositivi desktop.
Figure 11 Database page for desktop devices.

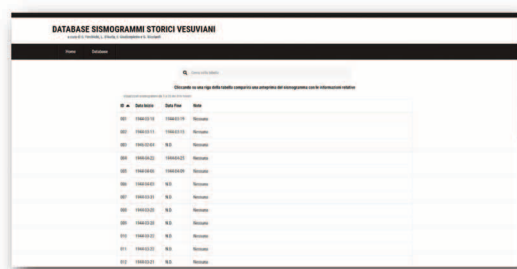
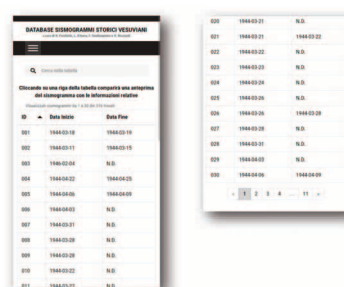


Figura 12 Database page per dispositivi mobile.
Figure 12 Database page for mobile devices.



2.2.2.1 Barra di Ricerca

Per consentire una più agile e mirata ricerca tra i circa 300 sismogrammi, è stata implementata una specifica funzionalità: ad essa si accede attraverso la barra di ricerca che consente di filtrare i sismogrammi da mostrare all'interno della sottostante tabella (Figura 13). Tale barra è caratterizzata da due pulsanti: una lente che permette di avviare la ricerca relativa al testo digitato; un pulsante a forma di "X" che permette di resettare la ricerca e ritornare alla visualizzazione iniziale della tabella con tutti i dati. All'interno della barra è possibile digitare la chiave della ricerca, che può essere: uno specifico anno, un ID, una parola da ricercare in una nota associata al sismogramma.



The screenshot shows a web interface titled "DATABASE SISMOGRAMMI STORICI VESUVIANI". It features a search bar at the top with a magnifying glass icon and a reset button (X). Below the search bar, a table displays search results. The table has columns for ID, Start Date, End Date, and Notes. The visible data is as follows:

ID	Data di Inizio	Data di Fine	Note
100	1980-01-01	1980-01-01	...
100	1980-01-01	1980-01-01	...
100	1980-01-01	1980-01-01	...
100	1980-01-01	1980-01-01	...
100	1980-01-01	1980-01-01	...
100	1980-01-01	1980-01-01	...

Figura 13 Esempio di ricerca per anno.

Figure 13 Search by year example.

Nel caso in cui la ricerca non conduca a sismogrammi esistenti, il portale restituisce la seguente schermata (Figura 14).

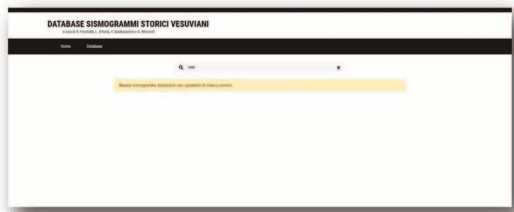


Figura 14 Output della ricerca nel caso in cui non siano stati trovati sismogrammi rispondenti alla chiave di ricerca.

Figure 14 Search output in case there are no seismograms for the search key.

Per i dettagli implementativi fare riferimento alla Sezione 4.4.

2.2.2.2 Tabella

La lista dei sismogrammi viene presentata attraverso una tabella che riporta una serie di informazioni (ID, Data di Inizio, Data di Fine, eventuali Note). Per la sua gestione è stato utilizzato un metodo in base al quale il numero di colonne cambia in relazione alla dimensione del dispositivo utilizzato: in pratica, il numero di colonne mostrate viene ridotto al diminuire della larghezza della finestra. Questo metodo non è sempre utilizzabile, perché quando le informazioni sono tutte fondamentali non è corretto nasconderele, ma, nello specifico caso, può essere adottato poiché la colonna che viene omessa è quella delle Note che molto spesso risulta essere vuota e in ogni caso tale informazione può essere visualizzata nella *preview* del sismogramma.

Di seguito le immagini della Tabella sia in caso di navigazione *desktop* (Figura 15) che *mobile* (Figura 16).

Figura 15 Tabella *desktop* con Note visibili.
Figure 15 Desktop table with visible Notes.

Visualizzati sismogrammi da 91 a 121 dei 316 trovati

ID ▲	Data Inizio	Data Fine	Note
091	1941-04-25	1941-04-27	Nessuna
092	1941-05-03	1941-05-07	Nessuna
093	1941-04-22	N.D.	Nessuna
094	1941-04-06	N.D.	Nessuna
095	1941-04-01	N.D.	Per riduzioni della lunghezza della penna (?) da cm 34 a cm 25, l'ingrandimento..._ridotto da l=170 a l=125
096	1941-03-19	N.D.	Nessuna

Figura 16 Tabella *mobile* con Note non visibili.
Figure 16 Mobile table with invisible Notes.

Visualizzati sismogrammi da 91 a 120 dei 316 trovati

ID ▲	Data Inizio	Data Fine
091	1941-04-25	1941-04-27
092	1941-05-03	1941-05-07
093	1941-04-22	N.D.
094	1941-04-06	N.D.

Per i dettagli implementativi fare riferimento alla Sezione 4.5.

2.2.2.2.1 Ordinamento della tabella

Un'altra funzionalità a disposizione dell'utente è quella di ordinamento delle informazioni della tabella in funzione del valore di una specifica colonna. In fase iniziale, la tabella riporta l'elenco dei sismogrammi organizzati in base al loro ID. L'utente ha la possibilità di cambiare l'ordinamento semplicemente cliccando sul *titolo* della colonna rispetto alla quale si preferisce visualizzare i dati: ID, Data inizio, Data fine.

Cliccando più volte sullo stesso titolo è anche possibile cambiare alternativamente la disposizione delle informazioni da crescente a decrescente e viceversa, rispetto al fattore selezionato.

Per i dettagli implementativi fare riferimento alla Sezione 4.5.2.

2.2.2.3 Barra di Paginazione

Si è convenuto di riportare all'interno della tabella un massimo di 30 sismogrammi per pagina. Ciò ha richiesto la creazione di una barra di Paginazione attraverso la quale potersi spostare tra i vari gruppi di sismogrammi.

Per implementare tale componente, è stata adottata una soluzione più articolata rispetto alle altre. Questa prevede che per i dispositivi *desktop*, i pulsanti associati alle pagine vengano mostrati tutti (Figura 17), mentre per i dispositivi *mobile* (Figura 18), vengano mostrati soltanto quattro pulsanti, intervallati da tre puntini "..." che indicano un salto tra le pagine.

Figura 17 Paginazione *desktop*.
Figure 17 Desktop paging.



Figura 18 Paginazione *mobile*.
Figure 18 Mobile paging.



Per i dettagli implementativi fare riferimento alla Sezioni 4.6 e 4.4.

2.3 Pagina *Preview*

Come già accennato, è possibile accedere ai dettagli di un sismogramma attraverso la pagina *Preview*.

La *preview* del sismogramma apre una nuova pagina semi-trasparente che si sovrappone e nasconde la pagina di partenza. Essa contiene un titolo, un'immagine della carta associata al sismogramma e le relative informazioni (Data Inizio, Data Fine e Note). È possibile chiudere la finestra della *preview* in due modi diversi. Il primo prevede un clic sul pulsante a forma di "X" che si trova vicino al titolo, mentre l'altro prevede un clic al di fuori dell'immagine (sulla parte scura).

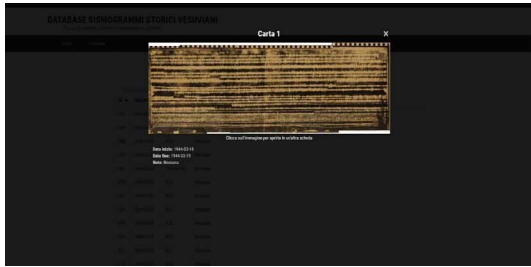


Figura 19 Pagina *Preview*.

Figure 19 Preview Page.

Per i dettagli implementativi fare riferimento alle Sezioni 4.6 e 4.7.

3. *Framework client-side* di sviluppo: *Bootstrap*

3.1 Alcuni concetti generali

Per approcciare alla progettazione di una pagina *web responsive*, bisogna procedere in maniera tale da:

- adattare il *layout* al più ampio numero di risoluzioni di schermo possibile;
- adattare le dimensioni delle immagini alla risoluzione e alle dimensioni dello schermo;
- semplificare il *layout* degli elementi presenti sulla pagina per i dispositivi mobili con schermi piccoli.

In generale, si possono seguire due approcci:

- *content-first*: tiene conto di ciò che la pagina *web* deve mostrare, cioè le informazioni. Questo concetto utilizza un approccio che mette al centro la gestione dei contenuti;
- *mobile-first*: utilizza un approccio che richiede che la progettazione del sito sia fatta a partire dalla strutturazione dei contenuti per i dispositivi con schermo a risoluzione più bassa, cioè per i dispositivi mobili.

Nello specifico, al fine ottenere una pagina *web responsive*, si può procedere attraverso l'utilizzo delle estensioni e delle proprietà *CSS* (*Cascading Style Sheets*). Il *CSS* è un linguaggio che gestisce il design e la presentazione delle pagine *web* (cioè l'aspetto estetico) e lavora in unione con l'*HTML*, o *HyperText Markup Language* (linguaggio a marcatori per ipertesti, il quale invece gestisce i contenuti delle pagine). Il *CSS* consente di specificare le regole secondo cui il sito deve presentare le informazioni, e allo stesso tempo, consente di tenere separati tutti i comandi relativi alla formattazione e allo stile (font, colori, dimensioni, ecc.) da quelli relativi al contenuto. L'estensione più importante di *CSS* a supporto della realizzazione di pagine *web responsive*, è quella delle *media queries*. Essa permette di usare diversi fogli di stile in base alle caratteristiche

del dispositivo utilizzato, permettendo l'introduzione di nuovi concetti come la *Resolution-Breakpoint*. Questo concetto prevede l'utilizzo di soglie in base alle quali modificare la presentazione grafica in funzione della larghezza del dispositivo. Solitamente vengono identificate quattro soglie:

- *Extra small*: per larghezza dispositivo inferiore a 768 pixel⁹.
- *Small*: per larghezza dispositivo dai 768 pixel ai 992 pixel.
- *Desktop*: per larghezza dispositivo dai 992 pixel ai 1200 pixel.
- *Large*: per larghezza dispositivo che va oltre i 1200 pixel.

Tra le proprietà del CSS più rilevanti per lo sviluppo di pagine *web responsive* troviamo quella del "CSS Flexbox" e quella del "CSS Grid System": entrambe le proprietà vengono utilizzate per organizzare componenti ed oggetti nella pagina secondo un *layout* predefinito (il container gioca il ruolo del padre che fornisce indicazioni ai figli su come disporsi nella pagina).

Un altro fattore da citare è la "viewport" utilizzata dai browser. La "viewport" è l'area di una pagina *web* visibile all'utente, quindi varia in base al dispositivo e sarà più piccola per i dispositivi mobili rispetto a quelli *desktop*. Prima dell'avvento di *tablet* e *smartphone* le pagine *web* erano progettate solo per schermi di computer, era quindi comune che esse avessero un design statico con una dimensione fissa. Però, con l'avvento dei dispositivi mobili, le pagine *web* di dimensioni fisse risultavano essere troppo grandi per adattarsi alla *viewport*. I browser hanno risolto questo problema riducendo l'intera pagina *web* per adattarla allo schermo. Questo non risolve però il problema: i contenuti possono diventare troppo piccoli per essere correttamente fruiti. Diventa quindi necessario prendere il controllo della *viewport* utilizzando un "meta" tag da inserire nella sezione "head" della pagina *HTML*. In questo modo ci si assicura che la larghezza della pagina sia pari alla larghezza del dispositivo e che non sia presente uno zoom iniziale. È possibile ottenere questo risultato con il seguente tag:

`"<meta name="viewport" content="width=devicewidth, initial-scale=1.0">"`.

3.2 Framework client-side

Strumenti e tecniche sopra citati hanno permesso di risolvere alcune problematiche, non forniscono però metodi per garantire la riusabilità del software. Per questo motivo entrano in gioco alcuni *framework client-side*, il cui scopo è quello di offrire agli sviluppatori *web* dei componenti o delle *utility* pronte all'uso per velocizzare e semplificare la fase di sviluppo. Ciò non solo migliora la riusabilità, ma rende il codice più leggibile e manutenibile. Questo è possibile perché durante lo sviluppo di pagine *web* moderne, vengono spesso utilizzati componenti e metodi di disposizione ricorrenti.

Nello specifico, i *client-side framework* sono composti da una serie di *file* che contengono strumenti utili a semplificare e velocizzare la realizzazione di una pagina *web*. Essi di solito sono formati da un *file CSS* e/o uno *script Javascript*. Il *file CSS* può fornire uno stile ai vari *tag HTML* o una serie di classi che si possono utilizzare per stilizzare le pagine e che aiutano a comporre dei componenti. Invece lo *script* può essere utile in molti modi diversi, per esempio può essere impiegato per definire componenti dinamici non realizzabili con solo *HTML* e *CSS*.

In sintesi, i *framework* sono caratterizzati da un sistema di *layout* a griglia *responsive*, un insieme di componenti ricorrenti ed un insieme di *CSS utilities* che consentono di velocizzare e semplificare la fase di sviluppo.

⁹ Il più piccolo elemento grafico visibile sullo schermo, corrispondente a un punto luminoso; ciascuno dei punti di cui si compone un'immagine digitale (dal Dizionario Corriere della Sera).

3.2.1 Bootstrap

Al fine di reingegnerizzare e di creare il nuovo portale *responsive* è stato utilizzato il *framework Bootstrap*. Esso è uno dei più popolari front-end *framework* e permette di realizzare velocemente siti *responsive* di tipo *Mobile-First*.

Tale *framework* mette a disposizione molti strumenti e componenti che permettono allo sviluppatore di risparmiare tempo prezioso, nello specifico:

- Un sistema di disposizione a griglie *responsive* (sezione 3.2.1.1).
- Un insieme completo di componenti (sezione 3.2.1.2).
- Un insieme di *CSS utilities*.
- Un insieme di *icone*.
- Un sistema di gestione della tipografia *web*.

Inoltre, offre un grande livello di compatibilità, supportando sia *browser mobile* che *desktop*. Il livello di supporto *browser* si estende anche per quelli *legacy* come *Internet Explorer* con versione superiore alla 10.

Di seguito la lista dei *browser* supportati: *Chrome* >=45; *Firefox* >= 38; *Edge* >= 12; *Explorer* >= 10; *iOS* >= 9; *Safari* >= 9; *Android* >= 4.4; *Opera* >= 30.

Per completezza, di seguito si descrivono le caratteristiche/componenti salienti del *framework*, alcune delle quali saranno utilizzate per lo sviluppo del nuovo portale.

Si precisa che se il lettore non è interessato agli aspetti di carattere implementativo, può tranquillamente tralasciare la lettura delle seguenti sottosezioni.

3.2.1.1 Layout

Per semplificare la disposizione delle varie componenti delle pagine *web*, *Bootstrap* fornisce: *containers* e *grid system*.

I *containers* sono l'elemento di *layout* più semplice e sono richiesti durante l'utilizzo della *grid system*. Il loro compito è quello di contenere le varie parti della pagina *web*, e di gestire la loro estensione in termini di larghezza. Per utilizzarli bisogna specificare il tipo di container attraverso l'attributo *class*.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
container	100%	540px	720px	960px	1140px
container-sm	100%	540px	720px	960px	1140px
container-md	100%	100%	720px	960px	1140px
container-lg	100%	100%	100%	960px	1140px
container-xl	100%	100%	100%	100%	1140px
container-fluid	100%	100%	100%	100%	100%

È anche presente un *container* che permette di specificare larghezza a 100% fino ad un determinato punto. Per usufruirne bisogna utilizzare la classe *container-{breakpoint}*, specificando in *breakpoint* il punto in cui si vuole che il contenuto non si espanda più in larghezza.

La *grid system*, invece, utilizza una serie di classi che sono di tipo *container*, *row* e *col* per disporre e allineare in maniera *responsive* il contenuto. Per fare ciò, viene utilizzata internamente una *feature* di *CSS* chiamata *flexbox*. In pratica bisogna utilizzare:

1. Un *container* per centrare e aggiungere *padding*¹⁰ orizzontale e per gestire la larghezza.
2. Delle *row* come figlie del *container* che rappresentano le righe della *grid system* ed hanno il compito di contenere le colonne.
3. Delle *colonne* come figlie di una *row* che individuano le singole celle ed hanno il compito di contenere le componenti di una cella. Esse presentano del *padding* orizzontale (chiamato *gutter*) che viene controbilanciato dal *margin* orizzontale negativo, in modo tale che il contenuto sia allineato a sinistra (per eliminare questo comportamento è possibile utilizzare la class *no-gutters* nella *row*).

Ogni riga può contenere 12 celle, ed ogni colonna può occupare una o più celle. Per specificare il numero di celle, viene utilizzata la classe *col-#{celle}*.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	auto	540px	720px	960px	1140px
Class prefix	col-	col-sm-	col-md-	col-lg-	col-xl-

Se non si specifica il numero di celle, tutte le colonne della riga assumono la stessa larghezza, ed è possibile adattare la dimensione di una colonna in base al suo contenuto, utilizzando la classe *col-{breakpoint}-auto*. *Bootstrap* mette a disposizione anche un metodo per specificare l'ordine delle colonne attraverso delle classi. Per fare ciò si possono utilizzare le classi *order-{numero}*.

3.2.1.2 Componenti

I componenti forniti da *Bootstrap* sono di tipo *responsive* e coprono una vasta gamma di necessità. Tra questi ne citiamo alcuni che sono stati utilizzati nel corso dell'implementazione del portale:

- **Table:** *Bootstrap* mette a disposizione delle classi per creare delle classiche tabelle in modo semplice e veloce. Bisogna aggiungere la classe *table* al tag *table*. È anche possibile specificare alcune caratteristiche aggiuntive aggiungendo al tag *table* una delle seguenti classi: *table-dark*, *table-striped*, *table-bordered*, *table-borderless*, *table-hover*, *table-sm*. Per permettere alle tabelle di essere *responsive* e quindi scrollabili orizzontalmente, bisogna contenere le tabelle in un tag *div* con classe *table-responsive*. Utilizzando le classi *table-responsive-{sm | md | lg | xl}* è possibile anche fare in modo che la tabella sia *responsive* fino ad un *breakpoint*.
- **Immagini:** *Bootstrap* fornisce una serie di classi per gestire le immagini e renderle *responsive*, facendo in modo che esse non diventino più grandi del loro genitore. Le classi principali sono: *img-fluid* che si occupa di scalare l'immagine insieme al suo genitore e *img-thumbnail* che consente di associare all'immagine un bordo arrotondato largo 1px.
- **Alert:** *Bootstrap* fornisce messaggi di *feedback* per azioni tipiche dell'utente, attraverso diversi tipi di *alert*. Questi *alert* sono disponibili con testi di qualsiasi lunghezza, ed è possibile aggiungere un bottone per la loro chiusura. Per utilizzarli basta aggiungere un tag *div* con classe *alert*. Esistono anche classi aggiuntive opzionali che permettono di

¹⁰ Tramite il *padding* viene definito uno spazio tra l'area dei contenuti ed il confine interno del box che lo contiene.

specificare il colore dell'*alert* e stilizzare gli elementi che vengono spesso aggiunti negli *alert* (per esempio i *link* con *tag a*¹¹ e classe *alert-link*).

- **Bottoni:** *Bootstrap* fornisce una serie di classi che consentono di usufruire di bottoni già stilizzati per varie necessità. Per utilizzarli bisogna creare un *tag button* con classe *btn*. A questo punto, in base al tipo di bottone di cui si necessita bisogna aggiungere una delle seguenti classi: *btn-primary*, *btn-secondary*, *btn-success*, *btn-danger*, *btn-warning*, *btn-info*, *btn-light*, *btn-dark*. Tali classi possono essere sostituite da delle controparti che forniscono lo stile per creare un *outline button*, aggiungendo la keyword *outline*. Sono disponibili anche altre classi aggiuntive: *text-nowrap* per assicurare che il testo non vada a capo, *btn-lg/btn-sm* per specificare la dimensione del bottone, *btn-block* per creare un bottone di tipo *block-level*, *active* per creare bottoni che appaiano già premuti.
- **Card:** *Bootstrap* fornisce delle *Card* che consentono di usufruire di contenitori flessibili ed estendibili con possibilità di aggiungere *header* e *footer*. Inoltre, forniscono strumenti per stilizzare tutto quello che può essere necessario in una *card*. Per utilizzarla bisogna creare un *tag <div>* con classe *card*. A questo punto alla *card* è possibile aggiungere (come figli) le varie sezioni che una *card* può avere: *immagine*, *header*, *body* e *footer*. Per la loro creazione si possono utilizzare le classi: *card-img-top*, *card-header*, *card-body* e *card-footer*. Per semplificare e velocizzare la creazione del contenuto di una *card*, è possibile usufruire anche delle classi: *card-title*, *card-subtitle*, *card-text*, *cardlink*. Nella documentazione ufficiale, vengono anche illustrati dei metodi per inserire una *nav* adattata alla *card*.
- **Carousel:** *Bootstrap* fornisce un componente che permette di creare una presentazione con slide cicliche. Nello specifico, mette a disposizione un metodo per creare presentazioni ed aggiungere eventuali controlli e indicatori per cambiare slide. Per usufruirne bisogna creare un elemento radice con determinati attributi, per poi aggiungere al suo interno le varie slide.
- **Dropdown:** permettono di attivare o disattivare delle sovrapposizioni utili per visualizzare elenchi di tasti o link. Il cambio di stato avviene tramite clic e non *hover*¹². Per utilizzarlo bisogna usare un *tag <div>* con classe *dropdown* ed aggiungere come figli un bottone con classe *dropdown-toggle* e un *tag <div>* con classe *dropdown-menu* che conterrà i link sotto forma di *tag <a>* con classe *dropdown-item*. Per stilizzare ulteriormente il *dropdown*, è possibile usare le classi: *dropdown-menu-right* per allineare a destra, *dropdown-header* per aggiungere un *header*, *dropdown-divider* per aggiungere una linea orizzontale che divide i gruppi di link.
- **Form:** sono componenti grafiche che permettono all'utente di interagire con il *web server* e ricevere eventuali servizi. In generale ogni *form* può accettare diversi tipi di elementi in *input*. *Bootstrap* offre un metodo per creare velocemente *form* già stilizzate attraverso l'utilizzo di classi che permettono la creazione dei vari elementi di *input*. Ogni elemento di *input* deve essere contenuto in un *tag div* con classe *form-group* e possono essere: *input* testuali o delle *select box* usando la classe *form-control*, dei file *input* usando la classe *form-control-file*, dei range *input* usando la classe *form-control-range*, delle *checkbox* o radio *button* usando la classe *form-check-input*. Nel caso in cui si necessita di avere *input* non modificabili, bisogna aggiungere l'attributo *readonly* all'elemento di *input*. Inoltre, è possibile anche mostrarlo come testo semplice, cambiando la classe dell'elemento da *form-control* a *form-control-plaintext*.
- **Modal:** Sono dei componenti utili per creare finestre di dialogo (delle *finestre modali*). Una prima caratteristica del *modal*¹³ è che esso si posiziona sempre sopra gli altri contenuti e

¹¹ Il tag *<a>* consente di collegare tramite link, altre pagine interne dello stesso sito o siti web esterni.

¹² *Hover* aggiunge uno stile ad un elemento quando il mouse è su di esso.

¹³ *Modal* è una finestra di dialogo/pop-up di dialogo che viene visualizzata sulla parte superiore della pagina corrente.

rimuove lo scroll dal *body* in maniera tale che esso possa essere applicato sul *modal*. Inoltre, supporta una sola finestra di dialogo alla volta. Per usare tale componente bisogna creare un tag `<div>` con classe pari a *modal* e attributo *role* pari a *dialog*. Al suo interno deve essere presente un altro tag `<div>` con classe *modal-dialog* che a sua volta deve contenere un tag `<div>` con classe *modal-content*. A questo punto è possibile aggiungere all'interno dell'ultimo tag tutte le parti del dialog, che possono essere: *header* attraverso la classe *modal-header*, *body* attraverso la classe *modal-body* e *footer* attraverso la classe *modal-footer*. Il *modal* creerà un *backdrop* (uno sfondo) che se viene cliccato ne determina la chiusura, a meno che non si aggiunga l'attributo *data-backdrop* pari a *static*. Per semplificare la gestione del *modal*, *Bootstrap* mette a disposizione metodi per aprire e chiudere la finestra modale e delle classi per stilizzare gli elementi *HTML* interni. Per aprire la *finestra modale*, è possibile usare un bottone aggiungendogli gli attributi *data-toggle* pari a *modal* e *data-target* pari ad un selettore *CSS* che permetta di selezionare il tag radice della *modal* (di solito si fornisce un ID al tag radice ed il bottone seleziona la *modal* attraverso quello specifico id). Invece per chiudere la *finestra modale* si può semplicemente aggiungere bottone con attributo *data-dismiss* pari a *modal*.

- *Nav*: sono dei componenti che consentono di ottenere una lista di link da utilizzare per la navigazione. Possono essere creati attraverso due strutture differenti: la prima prevede l'utilizzo di un tag `` con classe *nav* che contiene tag `` con classe *nav-item* (all'interno dei tag `` ci sono i link sottoforma di tag `<a>` con classe *nav-link*); la seconda, invece, prevede l'utilizzo di un tag `<nav>` con classe *nav* che contiene i vari link sottoforma di tag `<a>` con classe *nav-link*. Di default si ha una *nav* orizzontale allineata a sinistra, però è possibile cambiare il tipo di *nav* aggiungendo all'elemento radice la classe: *justify-content-center* per allineare al centro, *justify-content-end* per allineare a destra, *flex-column* per ottenere una *nav* verticale.
- *Spinner*: sono componenti che possono essere utilizzati per indicare lo stato di caricamento di un componente o della pagina stessa. Per utilizzarli basterà creare un tag `<div>` con classe *spinner-border* o *spinner-grow* e con *role* pari a *status*. È possibile indicare il colore dello *spinner* aggiungendo la classe *text-{primary | secondary | success | danger | warning | info | light | dark}*.

3.2.1.3 Altri Strumenti

Bootstrap offre anche molte altre classi utilizzabili nei vari elementi *HTML* per aggiungere delle proprietà *CSS* in modo veloce e semplice (per esempio: gestire il colore del testo, aggiungere margini o padding, ecc.). Inoltre, offre una libreria di icone, che sono state realizzate per funzionare coerentemente insieme ai vari componenti, ed un metodo per personalizzarne il tema.

4. Implementazione

In questa sezione saranno forniti alcuni dettagli relativi all'implementazione del portale. Molte delle classi di *Bootstrap* utilizzate e di seguito referenziate sono state già introdotte nella sezione 3.2.1, mentre nella sezione 2.2, sono state descritte le interfacce e le modalità di interazione. Di seguito saranno presentati i dettagli implementativi in maniera semplice e quanto più elementare possibile, anche se per una comprensione totale del codice si presuppone che il lettore abbia almeno una conoscenza di base di concetti quali *HTML* e *CSS*.

4.1 File e organizzazione in cartelle

Le *directory* con i *file* del portale sono state organizzate nel seguente modo.

Una *directory* principale con i file *HTML*, gli script *JavaScript* e le sotto-*directory styles* e *assets*:

- Nella cartella *styles* sono presenti tutti i fogli di stile *CSS* utilizzati.
- Nella cartella *assets* sono presenti tutte le immagini utilizzate per la costruzione del portale, tra cui anche le carte associate ai sismogrammi (le digitalizzazioni di oltre 300 sismogrammi storici vesuviani registrati nel periodo compreso tra il 1915 ed il 1944), che sono state organizzate in una *directory* interna in modo da semplificarne l'accesso. In considerazione del fatto che ogni sismogramma ha un identificativo numerico, il nome dell'immagine ad esso associata sarà del tipo *Carta {id}mini.png*. Tale scelta consente di accedere alle immagini con la sola conoscenza dell'identificativo del sismogramma.

4.2 Barra di Navigazione

Navbar e *sidenav* vengono utilizzate per navigare tra le pagine del portale.

Nella *navbar* si trova il nome del portale, i crediti ed in caso di dispositivo *desktop* un elenco orizzontale di link che fanno riferimento alle varie pagine, altrimenti verrà visualizzato un bottone che permette l'apertura della *sidenav*. A tale bottone è stata assegnata la classe *hidden-desktop* in maniera tale che essa non compaia nei dispositivi *desktop*.

La *sidenav*, invece, è essenzialmente una *modal* con al suo interno l'elenco verticale di link che fanno riferimento alle varie pagine ed un pulsante a forma di "x" che permette di chiudere la *modal* stessa.

Il titolo ed i crediti della *navbar* hanno la dimensione dei caratteri che varia in base alla larghezza della finestra del dispositivo, affinché il testo entri in un'unica riga. Per ottenere tale risultato sono state utilizzate le *CSS Media Query*. Per realizzare l'elenco orizzontale di link, è stato utilizzato il componente "nav" fornito da *Bootstrap*, aggiungendo la classe *hidden-mobile* per nascondere ai dispositivi mobili (Figura 20).

```

<ul class="site-nav nav hidden-mobile">
  <li class="nav-item">
    <a href="/" class="nav-link active site-link left-separator">Home</a>
  </li>
  <li class="nav-item">
    <a href="/database.html" class="nav-link site-link left-separator right-separator">Database</a>
  </li>
</ul>

```

Figura 20 Elenco orizzontale di link.

Figure 20 Horizontal links list.

Invece, per realizzare la *sidenav*, è stata utilizzata la *modal* fornita da *Bootstrap*, aggiungendo al suo interno una *aside* (Figura 21).

```

<div id="side-menu-modal" class="modal" tabindex="-1" role="dialog">
  <aside class="mobile-menu">
    <button type="button" class="close menu-close" data-dismiss="modal">
      <span aria-hidden="true">&times;</span>
    </button>
    <hr>
    <a href="/">Home</a>
    <hr>
    <a href="/database.html">Database</a>
    <hr>
  </aside>
</div>

```

Figura 21 Sidenav per i dispositivi mobili.

Figure 21 Sidenav for mobile devices.

È stato aggiunto un effetto *slide* all'apertura della *sidenav*, usando la classe "mobile-menu" dove sono definite le proprietà CSS:

- *transform* pari a "translateX(-100%)" cosicché essa si trovi alla sinistra della finestra;
- *transition* pari a "transform 250ms" cosicché appena viene modificata la *transform*, le modifiche appariranno come una animazione che dura 250 millisecondi. Inoltre, per fare in modo che funzioni, all'apertura della *modal* viene aggiunta la classe *translatex-0*, che applica la proprietà CSS *transform* pari a *translateX(0px)*.

4.3 Card

La *card* da realizzare è di tipo "orizzontale" per i dispositivi *desktop*, "verticale" per quelli *mobile* e contiene una immagine, un titolo ed un testo. Per ottenere questo risultato è stata utilizzata la *card* orizzontale fornita da *Bootstrap*, aggiungendo:

- un ID univoco ad ogni *card*;
- classe *bg-gray* per fornire lo sfondo grigio;
- un tag `<a>` che contiene l'immagine per permettere l'apertura della *preview*.

Il componente finale sarà del tipo (Figura 22).

Figura 22 Esempio di utilizzo della *card*.
Figure 22 Example of use of the card.

```
<div id="fig" class="card bg-gray my-5">
  <div class="row no-gutters">
    <div class="col-md-4">
      <a href="#" onclick="return openCardPreview('fig', 'N.D.', 'N.D.');">
        
      </a>
    </div>
    <div class="col-md-8">
      <div class="card-body">
        <h5 class="card-title font-weight-bold text-center">Titolo card</h5>
        <p class="card-text">Testo della card.</p>
      </div>
    </div>
  </div>
</div>
```

4.4 Barra di ricerca

Per l'implementazione della funzionalità di ricerca dei sismogrammi è stata utilizzata una form con un solo campo di tipo *text input*. Al suo interno è possibile digitare la chiave di ricerca. Le azioni vengono attivate attraverso un clic sulla lente di ingrandimento che avvia la ricerca relativa al testo digitato, oppure attraverso un pulsante a forma di "X" che permette di resettare la ricerca e il filtro (Figura 23).

Figura 23 Form per la ricerca.
Figure 23 Search form.

```
<form class="search-bar" action="/database.html" method="GET">
  <input name="keyword" type="text" class="search-input" placeholder="Cerca nella tabella">
  <a href="/database.html" class="search-bar-reset circular-btn d-none">
    <span>&times;</span>
  </a>
  <button type="submit" title="Ricerca" class="search-icon circular-btn">
    
  </button>
</form>
```

Quando vengono inoltrati i dati della *form*, la pagina viene reindirizzata sempre alla pagina contenente la tabella, ma con il parametro url *keyword* pari al filtro. Per implementarlo, è stato utilizzato direttamente il metodo *filter* all'array dei sismogrammi (Figura 24).


```

const filter = new URL(window.location.href).searchParams.get('keyword');
if (filter) {
  // nel caso in cui sia presente un filtro, esso verrà applicato ai dati
  const lowerCaseFilter = filter.toLowerCase();
  tableData = data.filter(rowData => {
    const { id, startTime, endTime, note } = rowData;
    const lowerCaseNote = note.toLowerCase();
    return (
      (('00' + id).slice(-3)) === (('00' + filter).slice(-3)) ||
      (startTime !== 'N.D.' && startTime.includes(filter)) ||
      (endTime !== 'N.D.' && endTime.includes(filter)) ||
      (lowerCaseNote !== 'nessuna' && lowerCaseNote.includes(lowerCaseFilter))
    );
  });
  document.querySelector('.search-bar .search-input').value = filter;
  // mostro il pulsante per X
  document.querySelector('.search-bar .search-bar-reset').classList.remove('d-none');
}

```

Figura 24 Filtro applicato dalla ricerca.

Figure 24 Search filter.

4.4.1 Alert

Questo componente viene mostrato quando una ricerca nella tabella non porta risultati. Nello specifico è stato utilizzato il componente fornito da *Bootstrap*, aggiungendo la classe *alert-warning* per fornire lo stile di cui si necessitava (Figura 25).

```

<ul class="site-nav nav hidden-mobile">
  <li class="nav-item">
    <a href="#" class="nav-link active site-link left-separator">Home/<a>
  </li>
  <li class="nav-item">
    <a href="/database.html" class="nav-link site-link left-separator right-separator">Database/<a>
  </li>
</ul>

```

Figura 25 Gestione Alert.

Figure 25 Alert management.

4.5 Tabella

La tabella, come abbiamo già visto nelle precedenti sezioni, elenca tutti i sismogrammi disponibili ed è caratterizzata da quattro colonne: ID, Data inizio, Data fine e Note. Attraverso l'evento *click* su una riga della tabella è possibile ottenere l'apertura della preview del sismogramma.

Per la gestione della tabella sono stati utilizzati il componente *table* fornito da *Bootstrap* e le classi *table-bordered* e *table-hover*.

La parte *responsive* della tabella prevede l'utilizzo di due tecniche non implementate nel componente fornito da *Bootstrap*. La prima evita la visualizzazione della colonna Note nei dispositivi mobili. Tale risultato è stato ottenuto aggiungendo la classe *hidden-mobile* a tutti gli elementi presenti in quella colonna. La seconda tecnica assicura che il testo delle Note (spesso molto lungo) occupi solo una riga e non vada oltre la larghezza della colonna. Per ottenere ciò, è stato utilizzato un metodo che permette di troncare il testo ed aggiungere tre puntini alla fine. Tale metodo prevede che l'elemento HTML che contiene il testo occupi tutta la larghezza della colonna e che abbia come proprietà CSS:

- *white-space* pari a *nowrap*;
- *overflow* pari a *hidden*;
- *text-overflow* pari a *ellipsis*.

Il codice HTML corrispondente viene riportato in Figura 26

Come è possibile notare dal codice, il corpo è inizialmente vuoto. Ciò è dovuto al fatto si è deciso di popolare la tabella attraverso codice JavaScript per rendere più semplice una futura modifica alla lista dei sismogrammi. Il popolamento avviene tramite l'utilizzo di una stringa JSON generata a partire dal codice di popolamento del vecchio *database*. Ciò permette di modificare la lista di sismogrammi attraverso una semplice modifica della stringa JSON (Figura 27).

Figura 26 HTML della tabella .
Figure 26 HTML of the table.

```
<table class="table table-bordered table-hover">
  <thead>
    <tr>
      <th name="id" id="id-col" scope="col" style="width: 7ch;" class="arrow-up">ID</th>
      <th name="startTime" id="start-time-col" scope="col" style="width: 14ch;">Data Inizio</th>
      <th name="endTime" id="end-time-col" scope="col" style="width: 13ch;">Data Fine</th>
      <th name="note" scope="col" class="hidden-mobile">Note</th>
    </tr>
  </thead>
  <tbody></tbody>
</table>
```

Figura 27 Codice associato al popolamento della tabella.
Figure 27 Code associated with the population of the table.

```
tableData.forEach((rowData) => {
  const tr = document.createElement('tr');
  // creazione della cella che conterrà il valore dell'id
  const idNode = document.createElement('td');
  idNode.textContent = ('00' + rowData.id).slice(-3); // trasforma il numero in stringa
  tr.appendChild(idNode);
  // creazione della cella che conterrà la data inizio
  const startTimeNode = document.createElement('td');
  startTimeNode.textContent = rowData.startTime;
  tr.appendChild(startTimeNode);
  // creazione della cella che conterrà la data fine
  const endTimeNode = document.createElement('td');
  endTimeNode.textContent = rowData.endTime;
  tr.appendChild(endTimeNode);
  // creazione della cella che conterrà le note
  const noteSpan = document.createElement('span');
  noteSpan.className = 'note';
  noteSpan.textContent = rowData.note;
  const noteNode = document.createElement('td');
  noteNode.className = 'hidden-mobile';
  noteNode.appendChild(noteSpan);
  tr.appendChild(noteNode);

  tbody.appendChild(tr);
  rows.push(tr);
});
```

Dopo aver popolato la tabella, per ogni riga, è stata registrata una *callback* da eseguire quando avviene l'evento *click* su una di esse. In questa *callback* si estraggono i dati della riga, che saranno utilizzati per mostrare la *Preview* (Figura 28).

Figura 28 Registrazione dei *callback* per l'apertura delle *preview*.
Figure 28 Registration of callbacks for opening previews.

```
document.querySelectorAll('.table tbody tr').forEach((tr) => {
  tr.addEventListener('click', () => {
    const idNode = tr.firstElementChild;
    const startTimeNode = idNode.nextElementSibling;
    const endTimeNode = startTimeNode.nextElementSibling;
    const noteSpan = endTimeNode.nextElementSibling.firstElementChild;

    openPreview(
      parseInt(idNode.textContent),
      startTimeNode.textContent,
      endTimeNode.textContent,
      noteSpan.textContent
    );
  });
});
```

4.5.1 Generazione dati in formato JSON

Come descritto poc'anzi, i dati dei vari sismogrammi sono stati generati a partire dal codice di popolamento del vecchio *database*. Questi dati sono presenti sotto forma di script SQL per un DBMS MySQL, dove è presente la creazione di una tabella e l'inserimento delle sue righe. Per ottenere la stringa JSON di cui c'era bisogno, si è deciso di implementare uno script con il linguaggio *python*. Questa scelta è dovuta dal fatto che *python* offre una libreria chiamata *sqlite3* che permette di utilizzare un *database* senza la necessità di avere un server in funzione: poiché però le sintassi di MySQL e SQLITE differiscono leggermente, si è reso necessario applicare alcune modifiche per poter eseguire lo script. Una volta fatto ciò è stato creato uno *script python* per popolare il *database*, per poi estrarre i dati e salvarli in un file in formato JSON (Figura 29).

```

<ul class="site-nav nav hidden-mobile">
  <li class="nav-item">
    <a href="/" class="nav-link active site-link left-separator">Home</a>
  </li>
  <li class="nav-item">
    <a href="/database.html" class="nav-link site-link left-separator right-separator">Database</a>
  </li>
</ul>

```

Figura 29 Script *python* per la generazione del JSON.
Figure 29 Python script for JSON generation.

4.5.2. Ordinamento della tabella

Un'altra funzionalità che non viene fornita da *Bootstrap* è la gestione dell'ordinamento delle righe della tabella in funzione del valore di una colonna. Di *default* le colonne della tabella vengono sempre presentate in ordine crescente rispetto all'ID del Sismogramma. Per modificare l'ordinamento è previsto che l'utente possa cliccare sulla cella che indica il nome della colonna (es. ID, Data inizio, Data fine). Se la tabella è già ordinata in funzione della colonna cliccata, l'ordinamento viene invertito (diventa decrescente), altrimenti la tabella sarà ordinata in ordine crescente rispetto al valore della colonna selezionata.

La funzionalità è stata implementata in modo da non dover cancellare le righe e riscriverle, ma semplicemente riordinarle. È stato anche aggiunto un indicatore di ordinamento, al fine di mostrare all'utente per quale colonna e in quale verso è stata ordinata la tabella. Per la gestione dell'indicatore sono state utilizzate le classi *arrow-up* per indicare l'ordinamento crescente e *arrow-down* per indicare l'ordinamento decrescente rispettivamente.

L'effettivo ordinamento della tabella è diviso in tre parti: l'ordinamento dell'array in funzione della colonna e del verso; l'aggiornamento delle righe della tabella; l'aggiornamento dell'indicatore di ordinamento (Figura 30).

```

switch(newOrder.column) {
  case 'id':
    if (newOrder.type === 'ASC') {
      data.sort((a, b) => a.id - b.id);
    } else {
      data.sort((a, b) => b.id - a.id);
    }
    break;
  case 'startTime': case 'endTime':
    const { column } = newOrder;
    if (newOrder.type === 'ASC') {
      data.sort((a, b) => _dateCompare(a[column], b[column]));
    } else {
      data.sort((a, b) => _dateCompare(b[column], a[column]));
    }
    break;
}

for (let i=0; i<data.length; i++)
  _updateTableRow(rows[i], data[i]);

// rimuove il vecchio l'indicatore di ordinamento
currCol.classList.remove('arrow-${table.order.type === 'ASC' ? 'up' : 'down'}');
// aggiunge il nuovo indicatore di ordinamento
nextCol.classList.add('arrow-${newOrder.type === 'ASC' ? 'up' : 'down'}');

```

Figura 30 Codice JS associato all'ordinamento della tabella.

Figure 30 JS code associated with the sorting of table items.

Per l'ordinamento, effettuato attraverso il confronto delle date, non è stato possibile utilizzare istanze della classe *Date* fornite da JavaScript, poiché non supportano date precedenti al primo gennaio del 1970. Per questo motivo è stata implementata anche una funzione per il confronto delle date (Figura 31).

Figura 31 Funzione che confronta due date (Javascript).

Figure 31 Function that compares two dates (Javascript).

```
function _dateCompare(date1, date2) {
  if (date1 === 'N.D.' && date2 === 'N.D.') {
    return 0;
  } else if (date1 !== 'N.D.' && date2 === 'N.D.') {
    return -1;
  } else if (date1 === 'N.D.' && date2 !== 'N.D.') {
    return 1;
  }

  let result;
  date1 = date1.split('-');
  date2 = date2.split('-');

  result = parseInt(date1[0]) - parseInt(date2[0]);
  if (result === 0) {
    result = parseInt(date1[1]) - parseInt(date2[1]);
    if (result === 0) {
      result = parseInt(date1[2]) - parseInt(date2[2]);
    }
  }
  return result;
}
```

Figura 32 Modifica di una specifica riga.

Figure 32 Changing a specification line.

```
function _updateTableRow(tr, rowData) {
  // aggiornamento dell'id
  const idNode = tr.firstElementChild;
  idNode.textContent = ('00' + rowData.id).slice(-3);
  // aggiornamento della data inizio
  const startTimeNode = idNode.nextElementSibling;
  startTimeNode.textContent = rowData.startTime;
  // aggiornamento della data fine
  const endTimeNode = startTimeNode.nextElementSibling;
  endTimeNode.textContent = rowData.endTime;
  // aggiornamento delle note
  const noteSpan = endTimeNode.nextElementSibling.firstElementChild;
  noteSpan.textContent = rowData.note;
}
```

4.6 Barra di Paginazione

La paginazione viene utilizzata per permettere la selezione della pagina della tabella. Per l'implementazione di tale funzionalità è stata utilizzata come base la componente *pagination* di *Bootstrap*, a cui sono state apportate delle modifiche per renderla *responsive* (Figura 33). Per la gestione della parte *mobile*, al fine di far rientrare tutto in una sola riga del dispositivo si è optato per una soluzione che mostra solo una parte delle pagine disponibili (Figura 34).

Figura 33 Paginazione *desktop*.

Figure 33 Desktop *pagination*.



Figura 34 Paginazione *mobile*.

Figure 34 Mobile *pagination*.



Al fine di consentire future modifiche o integrazioni al *database*, non è stata implementata una paginazione di tipo statico, ma connessa al numero di pagine. La sua creazione viene eseguita solo se ci sono più pagine e prevede l'inserimento dei pulsanti: pagina precedente, pagina successiva, pagina numerata e due pulsanti disabilitati che contengono dei puntini. Attualmente è stata utilizzata una paginazione che prevede 30 righe per pagina.

4.6.1 Selezione pagina

La selezione della pagina prevede l'aggiunta della classe *d-none* (nasconde l'elemento) fornita da *Bootstrap* a tutte le righe che non fanno parte della pagina corrente.

```
const offset = (pageNumber - 1) * rowsPerPage;
const nextOffset = offset + rowsPerPage;
const rows = document.querySelectorAll('.table tbody tr');

// mostro le righe che fanno parte della pagina e nascondo le altre
rows.forEach((row, index) => {
  row.className = index >= offset && index < nextOffset ? '' : 'd-none';
});

// deseleziono il bottone selezionato per selezionare quello della nuova pagina
const currPageItem = document.querySelector('.pagination .page-num.active');
currPageItem.classList.remove('active');
const pageLinks = document.querySelectorAll('.pagination .page-num');
pageLinks[pageNumber-1].classList.add('active');
```

Figura 35 Codice JS e classe *d-none*.

Figure 35 JS code and *d-none* class.

4.6.2 Gestione della parte *responsive*

Per l'implementazione della parte *responsive* della paginazione sono state usate le classi *hidden-desktop* e *hidden-mobile* al fine di gestire la presentazione o meno di pulsanti e puntini (Figura 36). L'algoritmo della paginazione *responsive* utilizzato prevede che per i dispositivi mobili:

- I pulsanti per la prima e ultima pagina vengano sempre mostrati.
- Il primo pulsante disabilitato con i puntini viene mostrato solo se il numero della pagina corrente è superiore a 3.
- Il secondo pulsante disabilitato con i puntini viene mostrato solo se il numero della pagina corrente è inferiore al numero totale di pagine - 3.
- Se il numero della pagina è minore o uguale a 2 allora vengono mostrati anche i pulsanti per le prime 4 pagine.
- Se il numero della pagina è maggiore o uguale al numero totale di pagine - 2 - allora vengono mostrati anche i pulsanti per le ultime 4 pagine.
- Se il numero della pagina è compreso tra 2 ed il numero totale di pagine - 2 - allora vengono mostrati i pulsanti della pagina corrente e della pagina successiva.

```
<nav aria-label="Navigazione tabella">
  <ul class="pagination justify-content-center">
    <li class="page-item prev-link">
      <button class="page-link pagination-btn"><span></span></button>
    </li>
    <li class="page-item page-num">
      <button class="page-link pagination-btn">1</button>
    </li>
    <li class="page-item dots disabled hidden-desktop">
      <button class="page-link pagination-btn">...</button>
    </li>
    <li class="page-item page-num hidden-mobile">
      <button class="page-link pagination-btn">2</button>
    </li>
    <li class="page-item page-num hidden-mobile">
      <button class="page-link pagination-btn">3</button>
    </li>
    <li class="page-item page-num active">
      <button class="page-link pagination-btn">4</button>
    </li>
    <li class="page-item page-num">
      <button class="page-link pagination-btn">5</button>
    </li>
    <li class="page-item page-num hidden-mobile">
      <button class="page-link pagination-btn">6</button>
    </li>
    <li class="page-item dots disabled hidden-desktop">
      <button class="page-link pagination-btn">...</button>
    </li>
    <li class="page-item page-num">
      <button class="page-link pagination-btn">7</button>
    </li>
    <li class="page-item next-link">
      <button class="page-link pagination-btn"><span></span></button>
    </li>
  </ul>
</nav>
```

Figura 36 Paginazione con 7 pagine totali e con pagina corrente pari a 4.

Figure 36 Pagination with 7 total pages and with current page equal to 4.

4.7 Preview

Per la *preview* dei sismogrammi o di una *card* si utilizza *modal*. Esso è un componente a comparsa che richiede anche l'utilizzo di codice JavaScript per modificare il contenuto della *preview* in base a ciò che si vuole mostrare. È stato utilizzato il componente *modal* fornito da *Bootstrap* per ottenere l'effetto comparsa e chiusura attraverso un bottone, per poi suddividere il contenuto nelle seguenti tre parti:

1. *header*: contiene il titolo ed il pulsante di chiusura della *finestra modale* (il titolo è racchiuso nel tag `<h3>` con classe `preview-title`, mentre il bottone di chiusura è rappresentato dal tag `<button>` con classe `preview-close`”).
2. *immagine*: mostra la carta associata al sismogramma. È racchiusa in un tag `<a>`, per dare la possibilità di aprire l'immagine in un'altra scheda del browser.
3. *corpo*: contiene le informazioni del sismogramma, sotto forma di Data Inizio, Data Fine ed eventuali Note.

Figura 37 HTML della Preview.
Figure 37 HTML of the Preview.

```
<div id="preview-modal" class="modal" tabindex="-1" role="dialog">
  <div class="container-sm preview-container">
    <div class="position-relative">
      <button type="button" class="close preview-close" data-dismiss="modal">
        <span aria-hidden="true">&times;</span>
      </button>
      <h3 class="preview-title mb-3 mx-5"></h3>
    </div>
    <div class="preview-link-wrapper">
      <a class="preview-link" href="" target="_blank">
        <img class="preview-image" src="" alt="Immagine di preview">
      </a>
    </div>
    <p class="preview-info mt-1">Clicca sull'immagine per aprirla in un'altra scheda</p>
    <div class="px-3 mt-3">
      <p class="preview-text"><strong>Data inizio: </strong><span></span></p>
      <p class="preview-text"><strong>Data fine: </strong><span></span></p>
      <p class="preview-text"><strong>Note: </strong><span></span></p>
      <br>
    </div>
  </div>
</div>
```

Come è possibile notare dal codice, l'URL (*Uniform Resource Locator*) dell'immagine ed i dati del sismogramma non vengono inseriti a priori, ma durante l'apertura della *finestra modale*. Per fare ciò sono state implementate due funzioni: una per aprire la *modale* con le informazioni di un sismogramma ed un'altra per aprire la *modale* con le informazioni di una *card* (Figura 37).

La prima funzione prevede che vengano passati come argomenti tutti i dati del sismogramma, che sono: ID, Data inizio, Data fine e Note.

Figura 38 Funzione per apertura della Preview.
Figure 38 Function to open Preview.

```
function openPreview(id, startTime, endTime, note) {
  // tutte le carte dei sismogrammi sono salvate sulla cartella assets/carte
  const hrefId = ('0000' + id).slice(-4);
  const href = `assets/carte/Carta ${hrefId}mini.png`;

  titleTag.textContent = `Carta ${id}`;
  linkTag.setAttribute('href', href);
  imgTag.setAttribute('src', href);
  dataTags[0].textContent = startTime;
  dataTags[1].textContent = endTime;
  dataTags[2].textContent = note;

  $('#preview-modal').modal('show');
}
```

La seconda funzione, invece, prevede che vengano passati come argomenti l'ID del componente *card*, Data Inizio e Data Fine. L'ID della *card* viene passato direttamente, perché nella *card* stessa sono contenute parte delle informazioni di cui si necessita.

In questa funzione viene restituito il valore *false* perché viene utilizzata come *callback* da eseguire come azione di eventi clic di tag <a>. Questa tecnica viene utilizzata nello standard HTML per prevenire il cambio di pagina (quando viene eseguita una callback).

```
function openCardPreview(cardId, dataInizio, dataFine) {
  const card = document.querySelector('#' + cardId);
  const title = card.querySelector('.card-title').textContent;
  const note = card.querySelector('.card-text').textContent;
  const href = card.querySelector('.card-img').getAttribute('src');

  titleTag.textContent = title;
  linkTag.className = 'preview-card-link';
  linkTag.setAttribute('href', href);
  imgTag.setAttribute('src', href);

  dataTags[0].textContent = dataInizio;
  dataTags[1].textContent = dataFine;
  dataTags[2].textContent = note;

  $('#preview-modal').modal('show');
  return false;
}
```

Figura 39 Funzione per apertura della *Preview* delle *card*.
Figure 39 Function to open card Preview.

5. Conclusioni

Il nuovo portale, realizzato attraverso un approccio di tipo *Responsive Web Design* e con il supporto del *framework Bootstrap*, è stato testato su vari dispositivi e su più *Browser* rispondendo sempre in maniera ottimale.

È stato anche effettuato un test su entrambi i portali (quello vecchio e quello *web responsive*) utilizzando *Google Search Console* [GSC, 2021], un tool che prende in input un sito web e consente di verificare:

- la facilità di utilizzo del sito su dispositivi mobili;
- se il sito è stato realizzato con *design mobile-friendly*;
- se l'architettura del sito è stata realizzata con accuratezza e nel pieno rispetto dei canoni basilari per una visualizzazione ottimale su supporti di piccola taglia.

Al fine di effettuare il test di ottimizzazione mobile è stato sufficiente fornire i due indirizzi *web*. La seguente figura mostra i risultati ottenuti.

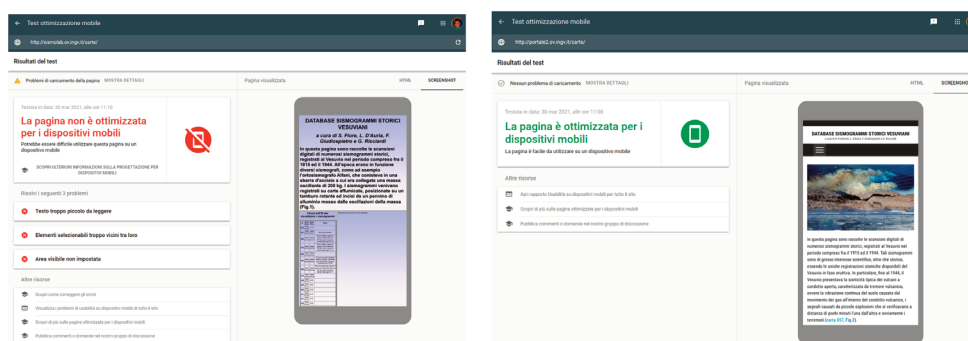


Figura 40 Risultati del Test Ottimizzazione *mobile*.
Figure 40 Mobile Optimization Test Results.

Nel caso del vecchio portale la pagina risulta non ottimizzata per i dispositivi mobili: il testo troppo piccolo da leggere, gli elementi selezionabili troppo vicini tra loro, l'area visibile non impostata.

Nel caso del nuovo portale la pagina risulta **Ottimizzata** per i dispositivi mobili e facile da usare. Un ulteriore ed essenziale miglioramento da evidenziare è quello relativo all'introduzione delle funzionalità di ricerca dei Sismogrammi. Se prima la ricerca poteva avvenire unicamente in maniera sequenziale (su 300 sismogrammi), ora è possibile effettuare sia la ricerca per parole chiavi (ID, Data, Nota) che modificare l'ordine (crescente o decrescente) delle righe della tabella sulla base dell'intestazione di una specifica colonna.

In sintesi, il nuovo portale:

- offre ai potenziali utenti un'esperienza di navigazione positiva, a prescindere dal dispositivo con cui si collegano, consentendo a chi naviga da smartphone di compiere qualsiasi azione in modo rapido ed efficace;
- semplifica la ricerca dei sismogrammi;
- consente di avere e mantenere un sito funzionale, leggero ed ottimizzato per i motori di ricerca.

Bibliografia

Fiore S., D'Auria L., Giudicepietro F., Caputo A., Ricciardi G., (2015). *Riproduzione digitale di sismogrammi storici del Vesuvio: prime registrazioni sismiche su un vulcano*. Rapporto Tecnico INGV, n. 294, pp. 18, ISSN 2039-7941, <https://doi.org/10.13127/rpt/294>

Sitografia

GSC, Google Search Console, (2021). <https://search.google.com/test/mobile-friendly?hl=IT>
Mark Otto e Jacob Thornton, Bootstrap, Framework for responsive site
<https://getbootstrap.com/>

QUADERNI di GEOFISICA

ISSN 1590-2595

<http://istituto.ingv.it/it/le-collane-editoriali-ingv/quaderni-di-geofisica.html/>

I QUADERNI DI GEOFISICA (QUAD. GEOFIS.) accolgono lavori, sia in italiano che in inglese, che diano particolare risalto alla pubblicazione di dati, misure, osservazioni e loro elaborazioni anche preliminari che necessitano di rapida diffusione nella comunità scientifica nazionale ed internazionale. Per questo scopo la pubblicazione on-line è particolarmente utile e fornisce accesso immediato a tutti i possibili utenti. Un Editorial Board multidisciplinare ed un accurato processo di peer-review garantiscono i requisiti di qualità per la pubblicazione dei contributi. I QUADERNI DI GEOFISICA sono presenti in "Emerging Sources Citation Index" di Clarivate Analytics, e in "Open Access Journals" di Scopus.

QUADERNI DI GEOFISICA (QUAD. GEOFIS.) welcome contributions, in Italian and/or in English, with special emphasis on preliminary elaborations of data, measures, and observations that need rapid and widespread diffusion in the scientific community. The on-line publication is particularly useful for this purpose, and a multidisciplinary Editorial Board with an accurate peer-review process provides the quality standard for the publication of the manuscripts. QUADERNI DI GEOFISICA are present in "Emerging Sources Citation Index" of Clarivate Analytics, and in "Open Access Journals" of Scopus.

RAPPORTI TECNICI INGV

ISSN 2039-7941

<http://istituto.ingv.it/it/le-collane-editoriali-ingv/rapporti-tecnici-ingv.html/>

I RAPPORTI TECNICI INGV (RAPP. TEC. INGV) pubblicano contributi, sia in italiano che in inglese, di tipo tecnologico come manuali, software, applicazioni ed innovazioni di strumentazioni, tecniche di raccolta dati di rilevante interesse tecnico-scientifico. I RAPPORTI TECNICI INGV sono pubblicati esclusivamente on-line per garantire agli autori rapidità di diffusione e agli utenti accesso immediato ai dati pubblicati. Un Editorial Board multidisciplinare ed un accurato processo di peer-review garantiscono i requisiti di qualità per la pubblicazione dei contributi.

RAPPORTI TECNICI INGV (RAPP. TEC. INGV) publish technological contributions (in Italian and/or in English) such as manuals, software, applications and implementations of instruments, and techniques of data collection. RAPPORTI TECNICI INGV are published online to guarantee celerity of diffusion and a prompt access to published data. A multidisciplinary Editorial Board and an accurate peer-review process provide the quality standard for the publication of the contributions.

MISCELLANEA INGV

ISSN 2039-6651

http://istituto.ingv.it/it/le-collane-editoriali-ingv/miscellanea-ingv.html

MISCELLANEA INGV (MISC. INGV) favorisce la pubblicazione di contributi scientifici riguardanti le attività svolte dall'INGV. In particolare, MISCELLANEA INGV raccoglie reports di progetti scientifici, proceedings di convegni, manuali, monografie di rilevante interesse, raccolte di articoli, ecc. La pubblicazione è esclusivamente on-line, completamente gratuita e garantisce tempi rapidi e grande diffusione sul web. L'Editorial Board INGV, grazie al suo carattere multidisciplinare, assicura i requisiti di qualità per la pubblicazione dei contributi sottomessi.

MISCELLANEA INGV (MISC. INGV) favours the publication of scientific contributions regarding the main activities carried out at INGV. In particular, MISCELLANEA INGV gathers reports of scientific projects, proceedings of meetings, manuals, relevant monographs, collections of articles etc. The journal is published online to guarantee celerity of diffusion on the internet. A multidisciplinary Editorial Board and an accurate peer-review process provide the quality standard for the publication of the contributions.

Coordinamento editoriale e impaginazione

Francesca DI STEFANO, Rossella CELI
Istituto Nazionale di Geofisica e Vulcanologia

Progetto grafico e impaginazione

Barbara ANGIONI
Istituto Nazionale di Geofisica e Vulcanologia

©2022

Istituto Nazionale di Geofisica e Vulcanologia
Via di Vigna Murata, 605
00143 Roma
tel. +39 06518601

www.ingv.it



Creative Commons Attribution 4.0 International (CC BY 4.0)

DATABASE SISMOGRAMMI STORICI VESUVIANI

a cura di G. Ferricksi, L. DiKato, F. DiStefano e G. Ricciardi

Home Database

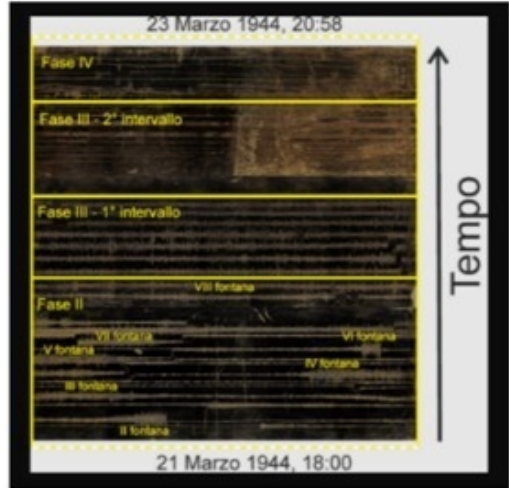
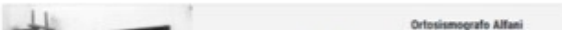


In questa pagina sono raccolte le scansioni digitali di numerosi sismogrammi storici registrati al Vesuvio nel periodo compreso fra il 1915 ed il grosso interesse scientifico, oltre che storico, essendo le uniche registrazioni sismiche disponibili del Vesuvio in fase eruttiva. In particolare, si tratta di sismogrammi registrati in condizioni aperte, caratterizzate da terreno vulcanico, ovvero la vibrazione continua del sasso causata dal movimento turbolento, il segnale causato da piccole esplosioni che si verificano a distanza di pochi minuti l'una dall'altra e ombreggia l'analisi.

Inoltre alcuni dei sismogrammi presenti nel database sono stati registrati proprio nel corso dell'eruzione del 1944, fornendo informazioni preziose sull'evoluzione del Vesuvio (parte 021, 022 e carta 025, Fig. 3).

Questi sismogrammi hanno consentito, ad esempio, lo studio delle fasi parossistiche dell'eruzione del Vesuvio del 1944, fornendo informazioni preziose durante le fasi eruttive più violente (Pappalardo et al., 2014).

Per maggiori dettagli: "Riproduzione digitale di sismogrammi storici del Vesuvio. Prime registrazioni sismiche su un vulcano" (Rapporti tecnici INGV 2015, numero 294).



DATABASE SISMOGRAMMI STORICI VESUVIANI

a cura di G. Ferricksi, L. DiKato, F. DiStefano e G. Ricciardi



Questi sismogrammi hanno consentito, ad esempio, lo studio delle fasi parossistiche dell'eruzione del Vesuvio del 1944, fornendo informazioni preziose sul comportamento del Vesuvio durante le fasi eruttive più violente (Pappalardo et al., 2014).

Per maggiori dettagli: "Riproduzione digitale di sismogrammi storici del Vesuvio: Prime registrazioni sismiche su un vulcano" (Rapporti tecnici INGV 2015, numero 294).



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA



Esempio di registrazione dell'attività sismica al Vesuvio

Tali scansioni si distinguono perché evidenziano che si sono registrate e ricorrono ad un periodo ben preciso.



In questa pagina sono raccolte le scansioni digitali di numerosi sismogrammi storici, registrati al Vesuvio nel periodo compreso fra il 1915 ed il 1944. Tali sismogrammi sono di grosso interesse scientifico, oltre che storico, essendo le uniche registrazioni sismiche disponibili del

