

4 Progettare un e-learning per il giapponese. Il case study *JaLea*

Sommario 4.1. Perché è importante l'experience design per l'e-learning? – 4.2 Perché è importante il Web 2.0 per l'e-learning? – 4.3 Strategie di embodiment in un prodotto e-learning. La learner experience. – 4.3.1 Il tempo nella relazione utente - artefatto digitale. – 4.3.2 Minimizzare i tempi di attesa. Tempo reale, percezione e tolleranza. – 4.3.3. Design collaborativo sostenibile: minimizzare i tempi di gestione e di inserimento contenuti. – 4.3.4 Sostenibilità e manutenibilità del software. Embodiment nel lungo periodo. – 4.3.5 Affordance e signifier nella pratica del Web design. – 4.3.6 Utenti e interfacce. – 4.4 Da BunpoHyDict a *JaLea*. – 4.5 *JaLea*, Your Japanese Learning System! – 4.5.1 *JaLea*: frontend e struttura. – 4.5.2 *JaLea*: proposte di utilizzo. – 4.5.3 *JaLea*: backend. – 4.6 *JaLea*: modalità di apprendimento attivo tramite esercizi. – 4.6.1 Motivazioni per la progettazione di un prototipo per l'apprendimento dei *kanji*. – 4.6.2 Realizzazione del prototipo.

Nel capitolo 3 sono state esposte le teorie alla base del processo di experience design e le caratteristiche principali di questo processo strettamente legato all'e-learning. In particolare, si ricorda:

- il fine delle strategie di progettazione dell'experience design, ovvero la creazione di un'esperienza ottimale per lo studente che gli permetta di relazionarsi in modo intuitivo e naturale con l'artefatto digitale;
- la multidisciplinarietà del processo di experience design che considera non solo la gestione e l'organizzazione dei contenuti (instructional design), ma anche la progettazione dell'interfaccia (interaction design), del sistema (system design), della grafica (graphic design), nonché l'utilizzo di strategie che permettano la manutenibilità del prodotto;

- l'identificazione di due aree di analisi differenti: front-office e back-office dedicate a utenti con competenze e obiettivi differenti.

Questo capitolo, attraverso l'osservazione del case study *JaLea*, descriverà il processo di creazione di strategie di experience design da più prospettive. Fondamentale è la prospettiva architettonica del sistema, tendenzialmente ignorata dalla letteratura di glottodidattica relativa all'e-learning, ma componente importante del processo di experience design. Pertanto, verranno presi in esame alcuni concetti chiave dal punto di vista tecnico di un'applicazione Web, applicati anche allo sviluppo di *JaLea*, permettendo di chiarire perché anche l'architettura del sistema è una componente importante da considerare nella progettazione di strumenti glottodidattici online. Questa analisi, inoltre, permetterà di contestualizzare i concetti di front-office e back-office e le strategie di experience design all'interno del case study attraverso la descrizione delle aree di frontend e backend.

4.1 Perché è importante l'experience design per l'e-learning?

Il processo di experience design applicato alla progettazione di un software e-learning ha il fine di agevolare il processo di interfacciamento dello studente con l'artefatto digitale. Se questo interfacciamento viene realizzato in modo ottimale, lo stato di 'flow' che si realizza permette la creazione inconscia di una relazione personale con l'artefatto il quale diventa parte del suo utilizzatore. La realizzazione di questo stato viene detto *embodiment*: «nel contesto di un'interazione di successo, l'artefatto diventa un'estensione del corpo di chi lo usa» (Triberti, Brivio 2016, 33). Sebbene la nascita della *embodied cognition* sia strettamente legata alle aspirazioni dell'Intelligenza Artificiale, ad oggi questo campo di studi è «del tutto emancipato dal costituire soltanto una risposta all'Intelligenza Artificiale e [...] costituisce un campo di studi autonomo e caratterizzato da un'alta varietà di approcci» (Triberti, Brivio 2016, 28). Tali approcci sono comunque accumulati da un visione della cognizione vicina alle teorie di Gibson. Egli infatti nel teorizzare gli oggetti caratterizzati da affordance (§ 3.1) sancisce la presenza di proprietà in ogni oggetto che promuove intrinsecamente una o più azioni, così come i nostri sensi ottici percepiscono forme e colori.

Un artefatto pertanto può fornire informazioni immediate a tal punto che gli strumenti utilizzati dall'uomo possano diventare parte del corpo del loro utilizzatore.

L'*embodiment* nei confronti di un artefatto digitale può rappresentare, secondo la teoria della mediazione di Riva e Mantovani (2012), un'azione mediata di secondo ordine.

Secondo questa teoria, a differenza di un'azione mediata di primo ordine rappresentata da un agente che controlla un artefatto prossimale (la racchetta, la mazza da golf), in un'azione mediata di secondo ordine, l'artefatto prossimale (la tastiera o il mouse) estende lo spazio corporale dell'utente al fine di controllare un artefatto distale (un avatar, un'interfaccia). In questo caso l'artefatto prossimale viene incorporato dall'utente e l'artefatto distale è oggetto di un processo di incarnazione.

Un'azione mediata di secondo ordine, che permette di sperimentare il processo di embodiment interamente attraverso l'incorporazione nell'artefatto prossimale e l'incarnazione nell'artefatto distale, è composta da una serie di passaggi che forniscono informazioni all'utente riguardo alla stessa attività mediata. Questi passaggi sono rappresentati dai vari elementi di cui l'artefatto complesso è composto, vale a dire nel caso del software e-learning, la struttura del sistema, l'interfaccia, le logiche di navigazione, la presentazione dei contenuti, l'aspetto grafico.

Quando l'embodiment si realizza, tali passaggi sono trasparenti, ovvero l'utente non si rende conto di essi e può in questo modo concentrarsi sull'azione fluida che si manifesta. È possibile però che tale processo non si realizzi, per una serie di problemi di natura diversa. Il caso più esemplare è la mancanza di feedback dai dispositivi coinvolti verso l'agente: nel caso di studio del presente lavoro potrebbero consistere ad esempio nella ricerca di un elemento grammaticale che indirizza verso una pagina senza alcun contenuto o messaggio d'errore, oppure nella pressione di un tasto dell'interfaccia che non permette di chiarire quale funzionalità si sia attivata. In questo caso il corto circuito generato interrompe l'effetto trasparenza (Winoograd, Flores 2008): «la tecnologia diventa manifesta e l'utente attiva processi cognitivi superiori per identificare la causa dell'errore» (Triberti, Brivio 2016, 40). Pertanto, è necessario che la progettazione dell'e-learning permetta di realizzare un prodotto che sia di aiuto, e non di interferenza, allo studente e questo può avvenire proprio grazie all'uso di strategie di experience design.

4.2 Perché è importante il Web 2.0 per l'e-learning?

Come già indicato nel capitolo 2, per 'applicazione Web' si intende un programma eseguibile tramite un browser che offre servizi più articolati e complessi di quelli informativi tipici dei siti Web 'vetrina'¹ e con una struttura (capitolo 3) che richiama per la forma dell'inter-

1 Il sito vetrina è un sito internet che viene utilizzato per presentare i prodotti e i servizi offerti dall'azienda rafforzando la sola percezione del brand aziendale. Non si parla quindi di un sito complesso con servizi personalizzati e sezioni di notizie, ma sem-

faccia quella delle applicazioni tipiche del Desktop. Sono ormai passati quasi 20 anni dalla nascita del Web 2.0,² ma questa evoluzione nell'ambito della programmazione Web è di fondamentale importanza per due aspetti principali: 1) il consolidamento della tecnologia che permette la divisione del codice dalle informazioni e 2) la possibilità di creare partecipazione attiva alla creazione dei contenuti: tramite questa tecnologia è possibile creare piattaforme attraverso le quali più persone possono confrontarsi e collaborare insieme.

Il prototipo alla base dell'applicativo *JaLea*, *BunpoHyDict* (Mariotti 2008), pur basato sui principi di apprendimento e-learning 2.0, interattivo, bottom-up, learner-centred (Downes 2012; Mariotti 2011; Tapscott 1998), era ancora legato al modello tipico del Web di tipo 1.0, in cui i contenuti testuali, quelli multimediali e i menu di navigazione erano inseriti direttamente tramite codice HTML. Le varie pagine erano copie di pagine precedenti che ne ripetevano quindi la struttura e l'impostazione grafica. Aggiornare i testi richiedeva di identificare il testo inserito fra il codice informatico, e prestare attenzione a intervenire solo su quello. Qualsiasi modifica all'organizzazione gerarchica delle informazioni richiedeva l'intervento su tutte le pagine precedentemente create. Con l'aumentare dei contenuti, il prodotto diventava sempre più difficile da gestire, e vista la complessità delle operazioni di modifica del testo, anche la formazione di eventuali nuovi collaboratori in grado di conoscere il giapponese e lavorare in HTML, richiedeva molto tempo.

Attraverso l'implementazione dei principi architetturali Web 2.0, invece, è possibile creare delle applicazioni dove chi desidera inserire nuovi contenuti, può farlo senza necessariamente possedere competenze tecniche o avere seguito un particolare corso di formazione.

Da un punto di vista di architettura di sistema, l'approccio più tradizionale per permettere la divisione delle informazioni dal codice (punto 1) vede l'implementazione di una struttura *client-server*.

Per essere visualizzata su un browser attraverso Internet, ovvero, qualsiasi pagina Web³ deve necessariamente risiedere all'interno di un server: un computer sempre connesso alla rete globale che offre il 'servizio' (da cui il nome server) di esposizione della pagina in un nodo della rete.

plicemente di un 'biglietto da visita' o brochure online. La prima tipologia di sito Web aziendale (<https://www.teutra.it/creazione-siti-Web/sito-vevtrina/>).

2 Termine coniato da Di Nucci nel 1999 e diffuso da O'Reilly nel 2004. Si veda anche O'Reilly 2007.

3 Si intendono le pagine Web remote, ovvero che non risiedono all'interno del computer dell'utilizzatore.

Tale servizio viene chiamato: servizio HTTP,⁴ dal nome del protocollo⁵ attraverso il quale il browser recepisce le informazioni della pagina Web. Il fatto che però il file risieda nel server non vuole necessariamente dire che il suo contenuto venga elaborato all'interno di questo. Ad esempio, le pagine cosiddette 'statiche' solitamente identificate dall'estensione '.html' vengono elaborate interamente dal browser e le informazioni, risultato di tale elaborazione, visualizzate all'interno di esso.

Al contrario delle pagine 'statiche', le pagine 'dinamiche' sono create con più linguaggi di programmazione tra cui uno o più linguaggi server-side (ovvero operanti all'interno del server). Nel server può essere installato anche un database che si occupa di contenere ed elaborare non il codice, ma i soli dati necessari all'applicativo, quali dati personali, di configurazione delle interfacce, dei menu e in generale i contenuti testuali e multimediali visualizzabili nell'applicazione.⁶ Quando il servizio HTTP elabora la pagina richiesta dal browser, verifica se ci sono porzioni di codice creato con linguaggi server-side all'interno di essa, e come prima cosa li elabora.

I linguaggi server-side vengono usati principalmente per definire le logiche di business, ovvero le regole e gli elementi legati alla visualizzazione delle informazioni o alla gestione della persistenza/memorizzazione dei dati. Tipici esempi di procedure scritte con codice server-side sono: chiamate al database o ad altri file presenti nel server, generazione di codice in HTML⁷ attraverso strutture condizionali che in base alla ricezione di determinati input permettono di definire come il programma si debba comportare. Questa logica, creata dagli algoritmi che definiscono le regole per il comportamento del programma in base agli input, viene chiamata: 'logica di business'.

4 In inglese HTTP (HyperText Transfer Protocol) server. Da notare che in questo caso 'server' si riferisce al servizio e non al computer connesso ad internet, condizione necessaria affinché la pagina Web sia consultabile dall'esterno.

5 Per una definizione di protocollo cf. *Enciclopedia Britannica*: <https://www.britanica.com/technology/protocol-computer-science>.

6 Un database non necessariamente risiede nello stesso server dove sono contenuti i file necessari all'applicativo.

7 Spesso è preferibile creare delle procedure che generino dinamicamente codice HTML partendo da contenuti variabili presenti nel database facendo uso dei linguaggi *server-side* i quali, a differenza di HTML che nasce primariamente come linguaggio di markup, includono caratteristiche propri dei linguaggi di programmazione.

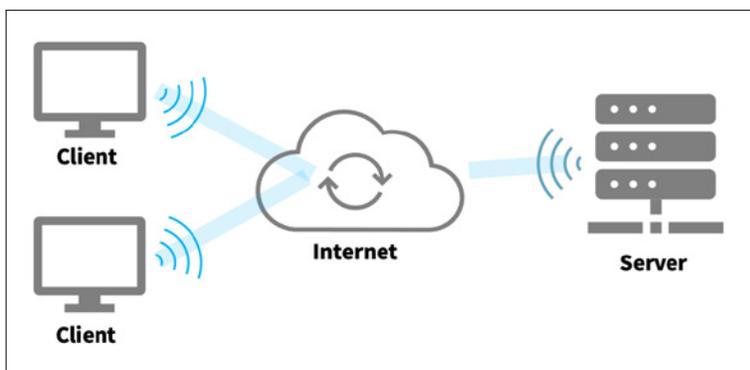


Figura 4.1 Schema di collegamento client-server

L'esempio più classico di una logica di business riguarda il modo con cui il software reagisce al momento dell'inserimento delle credenziali in un'area di autenticazione. Infatti, a seconda delle logiche di business implementate, l'inserimento corretto delle credenziali, permetterà di accedere a più funzionalità dell'applicativo o al contrario di utilizzarne solo poche o non potervi accedere affatto.

Altri esempi riguardano la profilazione dell'applicativo in base al paese di provenienza, o ancora, la presenza di funzionalità particolari che appaiono solo se prima sono stati completati uno più step quali la compilazione di un form di registrazione, di un sondaggio o la gestione della transazione con una carta di credito relativa all'acquisto di un bene.

La struttura architetturale client-server con un database rappresenta attualmente un modello particolarmente consolidato e comune nell'ambito delle applicazioni Web, tanto che, a seconda della tipologia di sistema operativo/software/database utilizzato, sono stati inventati degli acronimi per identificare l'ambiente rappresentato. Ad esempio, l'acronimo LAMP⁸ indica un applicativo Web presente su sistema operativo Linux,⁹ che utilizza Apache¹⁰ come servizio HTTP, MySQL¹¹ come database e PHP¹² come linguaggio installato nel server.

8 Wikipedia, «LAMP»: <https://it.wikipedia.org/w/index.php?title=LAMP>.

9 Wikipedia, «Linux»: <https://it.wikipedia.org/wiki/Linux>.

10 Wikipedia, «Apache»; «HTTP Server»: https://it.wikipedia.org/wiki/Apache_HTTP_Server.

11 Wikipedia, «MySQL»: <https://it.wikipedia.org/wiki/MySQL>.

12 Wikipedia, «PHP»: <https://it.wikipedia.org/wiki/PHP>.

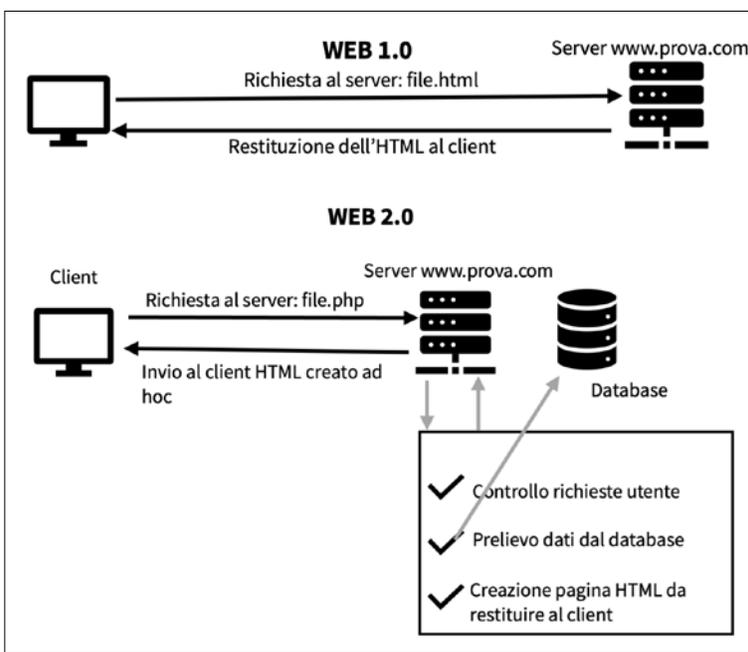


Figura 4.2 Differenza fra Web 1.0 e Web 2.0 dal punto di vista della comunicazione dei dati

Il secondo punto invece, la partecipazione, incoraggia l'utilizzo di paradigmi di divisione dell'applicativo in aree pubbliche e private, ovvero aree a cui l'accesso è garantito a tutti, o a molti, destinate principalmente all'utilizzo delle risorse dell'applicativo, distinte da aree in cui l'accesso è destinato a pochi, e che hanno lo scopo di permettere l'inserimento delle risorse quali ad esempio, contenuti e materiale multimediale.

Questo secondo punto (la partecipazione) è dipendente dal primo (la tecnologia): non è possibile creare una area riservata ad accesso limitato che permetta l'inserimento e la modifica di materiali multimediali, se non si utilizza una architettura dove i dati vengono memorizzati in un database o comunque trattati diversamente dal codice.

Di seguito, uno schema che esemplifica la differenza tra Web 1.0 e 2.0 dal punto di vista della gestione dei contenuti nella divisione area pubblica, destinata agli utenti finali e area privata, destinata agli amministratori. Questa divisione a livello di sviluppo del sistema e sviluppo dei contenuti realizza nella pratica il modello del processo di experience design introdotto nel capitolo precedente, in cui venivano identificati due tipi differenti di utenti: l'utente di back-office, che

possiede una serie di competenze lato corporation/università (creatore di contenuti, docente), e l'utente finale di front-office, che è il fruitore finale dei contenuti dell'applicativo (discente), inseriti e gestiti attraverso una area di amministrazione dedicata (back-office).

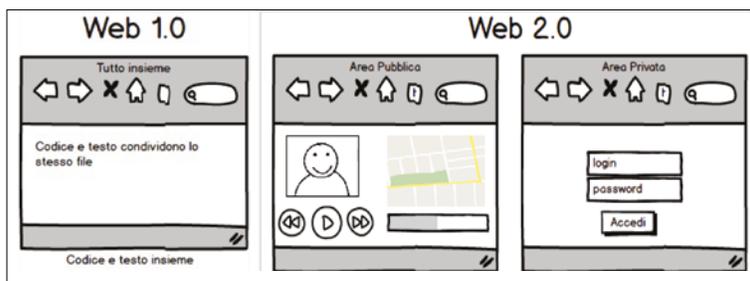


Figura 4.3 Differenza fra Web 1.0 e Web 2.0 dal punto di vista della divisione delle aree di competenza

Con il Web 2.0, in pratica, si realizza la possibilità di dividere nettamente in due le competenze dei soggetti partecipi allo sviluppo dell'applicativo. La figura dello sviluppatore si divide da quella del progettista di contenuti (content manager). Il primo si occuperà principalmente dello sviluppo dell'applicativo dal punto di vista tecnologico, il secondo dell'ideazione, dell'inserimento e dell'organizzazione del materiale. Anche se è auspicabile la comunicazione tra le due figure per la buona riuscita del progetto, esse sono comunque libere di lavorare in libertà e con i propri tempi.

La realizzazione della filosofia del Web 2.0, la trasformazione cioè del sito Web in piattaforma/applicazione, richiede anche una riorganizzazione del workflow di sviluppo. Se prima, con il Web 1.0, era sufficiente dare spazio al testo e ai contenuti multimediali, ora ogni singola fase di sviluppo, architettura di sistema, logiche di business, interfaccia, grafica, viene analizzata con cura in quanto il progetto è composto da un'architettura più complessa e delicata, strettamente interlacciata in ogni minimo ingranaggio di sistema. È per tale motivo che sviluppatori ed aziende, così come committenti legati alla didattica online, riconoscono l'importanza e la necessità di seguire un processo di experience design per lo sviluppo delle applicazioni Web.

4.3 Strategie di embodiment in un prodotto e-learning. La learner experience

Dopo avere delineato come funziona la tecnologia alla base del Web 2.0, si identificano alcune strategie fondamentali che, se implementate correttamente, concorrono alla realizzazione di uno stato di embodiment tra l'utente (il discente) e l'artefatto digitale (applicazioni e-learning).

Nel caso di un e-learning per l'apprendimento del giapponese, ad esempio, di particolare importanza sono i contenuti, realizzati attraverso best practices di experience design (si vedano i principi di Mayer nel § 1.5) che devono considerare le problematiche intrinseche del giapponese, come la necessità di molteplici trascrizioni (*furigana* e *rōmaji*) per venire incontro alle esigenze di studenti con competenze linguistiche differenti. Dall'analisi dello stato dell'arte dal capitolo 2, inoltre, è auspicabile l'uso di contenuti autentici e di materiali multimediali distribuiti con formati compatibili con i browser contemporanei e facilmente aggiornabili nel tempo.

Oltre ai contenuti, sono fondamentali:

- a. la struttura informatica dell'applicativo e i processi funzionali dell'interfaccia, realizzate attraverso strategie di System Development e Interaction design;
- b. l'aspetto grafico dell'interfaccia e la sua adattabilità a device differenti.¹³

Relativamente alla struttura dell'applicativo e dell'interfaccia (punto a), ai fini della realizzazione del processo di embodiment, essa deve considerare due aspetti: la minimizzazione dei tempi di interfacciamento utente-artefatto, e il livello di sostenibilità e manutenibilità dell'applicativo. Entrambi questi aspetti sono legati al concetto del tempo.

Il primo aspetto, la minimizzazione dei tempi di interfacciamento con l'utente, si lega alle nuove necessità dei giovani della generazione Z, discusse nel capitolo 2.

4.3.1 Il tempo nella relazione utente - artefatto digitale

Come sostiene Seow (2008), «la percezione è tutto»: maggiore è la discrepanza tra il tempo percepito dall'agente per la realizzazione del compito, e maggiore è la possibilità che la tecnologia diventi manifesta, impedendo la realizzazione del flusso (flow) e della conseguente incorporazione (embodiment) trasparente con il sistema.

¹³ Il concetto di responsività, ovvero adattabilità a device differenti è stato introdotto nel capitolo 1.

Egli, in particolare, introduce due concetti fondamentali riguardo alla relazione tra l'utente e l'artefatto distale: percezione e tolleranza. La percezione è il tempo registrato dalla nostra mente per il completamento di un compito. Questo tempo differisce da quello che ci vuole effettivamente per il completamento di tale compito. La tolleranza è invece il parametro che indica il limite entro il quale il tempo d'esecuzione di un determinato compito risulta accettabile.

Numerose ipotesi sulla percezione del tempo da parte del nostro cervello, come quella di Church, Meck e Gibbon (Church et al. 1994, *scalar timing theory*) vedono l'esistenza di una memoria di riferimento che memorizza i tempi di durata delle azioni. I tempi memorizzati nella memoria di riferimento vengono paragonati con quelli percepiti in tempo reale nella memoria di lavoro, e in base a questo viene calcolato il tempo di tolleranza.

Pertanto, la strategia nei confronti della gestione dei tempi di risposta dell'applicativo è triplice. Prendendo ad esempio lo scaricamento di un documento dalla rete, si possono ipotizzare strategie che considerano le seguenti problematiche:

- come fare in modo che il download sia più veloce? (Tempo reale);
- come fare in modo che il download sembri più veloce? (Percezione);
- come fare in modo che gli utenti tollerino il download? (Tolleranza).¹⁴

Considerando la relazione discente-e-learning, la questione da affrontare sarà la seguente: come incrementare i tempi di risposta del programma? Come fare in modo che la fruizione degli elementi sembri più veloce? Come aumentare il grado di tolleranza dei tempi di attesa?

4.3.2 Minimizzare i tempi di attesa. Tempo reale, percezione e tolleranza

Dal punto di vista dell'ingegneria delle applicazioni Web, le tecniche di solito utilizzate per migliorare le prestazioni dell'applicativo riguardano l'ottimizzazione delle query (interrogazioni al database tramite un particolare linguaggio per ottenere i dati desiderati) di modo da ridurre i tempi in cui i dati si trasferiscono al programma. Questa strategia, tuttavia, interviene solo sulla diminuzione del tempo reale necessario per spostare i dati da un punto (il database) all'altro (il programma). Esiste invece una tecnica relativamente moderna, che permette di cambiare il paradigma di comunicazione client-server degli anni Novanta e intervenire anche sulla percezione e sulla tolleranza del tempo da parte dell'utente, e di conseguenza rendere più compiuto l'embodiment con l'artefatto digitale, nel caso di studio della presente monografia quindi, del discente con il sistema e-learning.

¹⁴ Per ulteriori dettagli su percezione e tolleranza si veda Seow 2008.

La modalità tradizionale di comunicazione tra il client e il server, infatti, prevede che ogni volta che attraverso l'interfaccia si chiami una nuova voce di menu o una nuova funzionalità, il client ricarichi totalmente una nuova pagina dal server. Anche un'attività quale visualizzare un nuovo tasto, o un nuovo form da compilare, richiede cioè che tutta la pagina venga ricaricata, e non solo la nuova porzione dell'interfaccia che si intende visualizzare. Durante questa operazione la pagina si blocca e pertanto, l'utente avverte una sensazione di stasi nei confronti della quale è inerme.

Per ovviare alla lentezza del passaggio di informazioni sincrono da server a client, nel 2005 è stata sviluppata una tecnica chiamata AJAX (Garrett 2005). Sebbene esistessero implementazioni sperimentali prima di allora, fu solo nel 2004 e 2005 che la relativa implementazione divenne una *best practice*, anche grazie allo sviluppo dell'applicazione Google Maps che ne fece largo uso (Kawamata 2007).

AJAX è il sinonimo di Asynchronous JavaScript and XML; attraverso questa tecnologia è possibile, utilizzando specifici comandi in JavaScript (linguaggio di programmazione presente nel browser), ricevere dal server dati generici o in HTML e inserirli in una posizione specifica della pagina, evitando il caricamento intero di questa. Lo schema seguente riassume la differenza tra la modalità di comunicazione *client-server* sincrona, e una comunicazione mediata tramite AJAX.

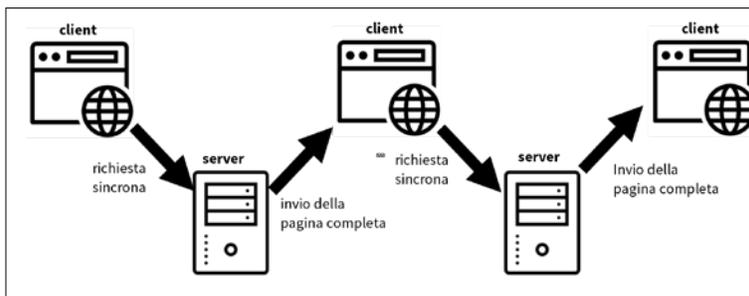


Figura 4.4 Modalità tradizionale di comunicazione client-server

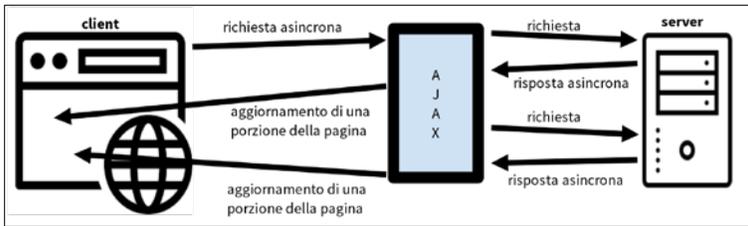


Figura 4.5 Modalità di comunicazione client-server tramite AJAX: la richiesta verso il server non implica il ricaricamento della pagina. Una volta ottenuta la risposta dal server viene aggiornata una porzione della pagina

L'utilizzo di AJAX permette, inoltre, non solo la fruizione delle varie componenti dell'applicativo in tempi più rapidi, ma la realizzazione di interfacce per la comunicazione utente-sistema quali chat, mappe e grafici modificabili e aggiornabili in tempo reale. Forse alcune di queste funzionalità non sembrano particolarmente innovative, visto che sono di uso comune nei programmi offline installabili nei sistemi operativi, ma per applicazioni Web, prima della nascita di AJAX, era impossibile realizzare sistemi di interazione in tempo reale se non attraverso componenti aggiuntivi esterni come Adobe Flash che attualmente è deprecato.¹⁵

Grazie alla richiesta al server e relativo caricamento di una sola porzione di pagina, AJAX porta a diminuire i tempi di caricamento (tempo reale). Poiché la pagina non viene ricaricata, infatti, l'interfaccia continua ad essere fruibile normalmente, permettendo all'utente di continuare ad interagire con l'artefatto digitale, diminuendo così anche la percezione del tempo impiegato per l'aggiornamento della pagina, e mantenendo ininterrotto il flusso di incorporazione (embodiment) discente-sistema e-learning,

Nel caso debba essere modificata l'interfaccia stessa, proprio mentre è utilizzata dall'utente, una pratica comune per impedire momentaneamente l'accesso all'utente è quella di posizionare uno strato (*layer*) trasparente sulla finestra aperta del browser con un circolo rotante (*spinner*). In questo modo l'utente viene visivamente avvisato con un signifier (Norman 2013) che il sistema sta attuando un caricamento dati. Questa pratica permette di aumentare la tolleranza dell'utente nei confronti dei tempi necessari per l'aggiornamento.

¹⁵ Per deprecato si intende non più in uso o ufficialmente supportato. Per dettagli su Adobe Flash si veda il § 2.1.



Figura 4.6 Istantanea di una pagina di *JaLea* durante il caricamento di contenuti tramite AJAX. Notare lo spinner centrale che indica l'attesa del caricamento

4.3.3 Design collaborativo sostenibile: minimizzare i tempi di gestione e di inserimento contenuti

In questo paragrafo si tratteranno gli aspetti legati alla creazione di un'area back-office (backend) per l'inserimento e la gestione dei contenuti in un sistema e-learning. La divisione in aree di backend e frontend (§ 4.5.1) dell'applicativo implica, come suggerito anche nell'introduzione, che tali aree verranno utilizzate da soggetti con fini e necessità differenti. Nel caso dell'area di backend, normalmente gli utenti sono quelli appartenenti a un gruppo o organizzazione legato alla gestione creativa e amministrativa dei materiali erogati tramite il software (§ 3.5).

L'area di backend infatti è fondamentale per permettere la creazione collaborativa dei contenuti. È necessario pertanto che possa permettere a più utenti di collegarsi con credenziali diverse. Al fine di ridurre i tempi necessari a creare le utenze, conviene predisporre una funzionalità apposita proprio all'interno del backend stesso.

I collaboratori che accedono a quest'area possono essere docenti, esperti della materia, ma anche studenti, tutti partecipanti alla costruzione della conoscenza. Questo approccio, chiamato design collaborativo, è un elemento fondamentale di sostenibilità del sistema perché incide su due dei tre pilastri (ambientale, sociale ed economico) del modello di sviluppo sostenibile proposto da René Passet (Kongoli 2016): quello sociale e quello economico.

Il design partecipativo può aumentare sia la sostenibilità sociale (aumentando l'accettabilità del progetto) che quella economica (aumentando le probabilità di successo del progetto). (Busson 2016, 19)

In questo caso le strategie da identificare sono duplici: 1) l'interfaccia deve comunque sottostare ad un processo di experience design che consenta all'utente di relazionarsi con essa (embodiment); 2) e al contempo deve facilitare il processo di creazione dei contenuti dell'utente.

I tre principi di Platt (2016) infatti, come suggerito anche da Ryan e Deci (2000, self-determination theory), identificano la motivazione per l'utilizzo del software non solo nel piacere, ma anche nell'utilità:

1. il valore del software si misura nel grado in cui può rendere un utente felice o produttivo, il software non ha alcun valore di per sé;
2. il software aumenta la produttività o la felicità dell'utente in 2 modi: a) aiuta a risolvere un problema specifico quale scrivere un articolo, pagare una bolletta, guidare una macchina; b) offre all'utente ambienti piacevoli, in quanto lo mette in condizione, ad esempio, di ascoltare la musica, giocare a un videogame, comunicare con una persona cara. Questi due punti, utilità e senso di piacevolezza, in alcuni casi possono essere entrambi presenti in grado differente;
3. in nessuna delle attività dei due casi a) e b) del punto precedente, l'utente medio vuole riflettere su come sta utilizzando il sistema. Mai. Nel primo caso l'utente è concentrato sul problema da risolvere, nel secondo caso, l'utente vuole mantenere il proprio stato di piacere. Qualsiasi elemento di disturbo o di distrazione, nel caso b), viene probabilmente percepito in modo più fastidioso a un'interruzione di un'attività lavorativa.

Questi principi inducono a pensare che se il processo di creazione dei contenuti è troppo complesso e lungo, l'utente del backend (quindi, di solito, non il discente) è meno motivato a collaborare e perde interesse nell'utilizzo del software. Proprio per questo è necessario identificare anche strategie sostenibili per l'inserimento e la gestione dei materiali. Il § 4.5.3 è dedicato proprio a questo argomento relativamente all'inserimento dei contenuti in *JaLea*.

4.3.4 Sostenibilità e manutenibilità del software. Embodiment nel lungo periodo

Per permettere all'utente di utilizzare l'applicativo nel lungo periodo, è necessario applicare una serie di strategie in ambito di progettazione e sviluppo del software che ne permettano la sostenibilità e la manutenibilità, sia a livello economico che a livello ambientale, sociale e di tempo.

La sostenibilità economica mira a portare ai massimi livelli il profitto ottenibile dallo sviluppo dell'applicativo, profitto da non intendersi solamente in termini finanziari, in quanto, come indicato nel paragrafo precedente, il valore del software è dato unicamente dal valore conferito ad esso dai suoi utenti. Pertanto, la sostenibilità economica è la capacità di sviluppare l'applicazione e i contenuti, non solo senza eccedere dal budget a disposizione, ma anche facendo in modo che gli utenti siano motivati ad utilizzarlo per un lungo periodo di tempo. In questo contesto si identificherà la manutenibilità come capacità di garantire nel minor tempo possibile:

- a. *bug fixing* (risoluzione di problemi tecnici) ed estensioni di funzionalità da parte dello sviluppatore;
- b. inserimento e modifica del materiale.

Sostenibilità e manutenibilità pertanto, sono relative non solo alla struttura del software, ma anche al processo di inserimento dei contenuti.

Relativamente alla gestione dello sviluppo del software, i punti da considerare per garantire sostenibilità e manutenibilità sono molteplici. Da un punto di vista prettamente legato allo sviluppo dell'applicativo¹⁶ i costi da tenere in considerazione sono:

1. costi di software e licenze correlate;
2. tempi di sviluppo.

È possibile inoltre identificare almeno tre fasi legate allo sviluppo:

1. la pianificazione;
2. lo sviluppo del prodotto attraverso la creazione del codice;
3. il supporto: bug fixing e nuove implementazioni.

Nella fase di pianificazione si analizzano le tecnologie da utilizzare in base ai requisiti di mercato (*business requirements*) del prodotto. Questa fase è la principale in quanto tutte le decisioni prese influenzeranno i tempi di sviluppo del punto 2 e del punto 3. In questa fase si analizzano anche i costi delle licenze necessarie per i software

16 Non si considerano quindi i costi legati all'hardware utilizzato, né i costi legati alla gestione aziendale che fa da supporto allo sviluppo.

legati allo sviluppo, o che possono essere di integrazione al prodotto da sviluppare.

La fase 2 (sviluppo) riguarda anche l'identificazione e l'utilizzo di un framework utile all'organizzazione del codice, in modo che sia più facile da gestire per gli utenti. Nel caso di *JaLea* è stato utilizzato un framework per PHP chiamato *CodeIgniter*¹⁷ che permette di organizzare il codice secondo il pattern architetturale MVC (*Model View Controller*) in grado di separare fundamentalmente la presentazione dei dati dai parametri ed algoritmi che creano le condizioni affinché i dati stessi vengano presentati (logica di business). Applicato allo sviluppo di un'applicazione Web, il pattern MVC permette di identificare delle aree di visualizzazione (*view*), che contengono i dati relativi alla sola rappresentazione dell'interfaccia utente attraverso il codice HTML, e delle aree di controllo (*controller*) che identificano le logiche di business. La terza area, *Model*, fornisce i metodi per accedere ai dati contenuti nel database. Pertanto, conterrà comandi per interrogare il database e restituire i dati richiesti. Per ottimizzare i tempi di sviluppo e la presenza di eventuali errori di funzionamento del sistema (bug), durante questa fase è necessario valutare l'utilizzo di librerie esterne, ovvero blocchi di codice già pronti che riguardano funzionalità che si desidera implementare.

La fase 3 (supporto) riguarda la post-produzione: una volta terminata la programmazione dell'applicativo o di una porzione di esso, si verificano eventuali errori, si procede a correggerli (bug fixing) e si valuta l'implementazione di nuove funzionalità. Se il codice è coerente e organizzato (ad esempio come il framework MVC descritto al punto 2), sarà più facile identificare eventuali errori e implementare nuove funzioni, anche dopo un possibile periodo di pausa dall'attività di programmazione, prima dell'inizio della tappa successiva del progetto (milestone).¹⁸

In fase di progettazione, inoltre al fine di garantire la manutenibilità del software e valutarne i costi di sviluppo, vengono considerati tre punti: il possibile utilizzo di software libero, temi e layout, librerie ed API disponibili.

¹⁷ Si veda la pagina del progetto: <https://codeigniter.com>.

¹⁸ Il termine *milestone* indica uno dei vari traguardi pianificati in cui è possibile suddividere l'attività di sviluppo di un progetto. Con una pausa prolungata dall'attività, è possibile dimenticare alcune delle complicate logiche di business che costituiscono gli algoritmi, ovvero l'insieme di regole condizionali per il funzionamento di un sistema. Un framework solido che garantisce il rispetto delle best practices, può essere molto utile, in questo caso, per riprendere in poco tempo familiarità con le logiche del progetto temporaneamente sospeso.

1) Verifica del possibile utilizzo di software libero

Il 'software libero' è rilasciato con licenze che ne permettono la libera esecuzione e copia, nonché l'analisi e la modifica del codice. Solitamente il software libero è rilasciato gratuitamente. Essendo possibile esaminare il codice, si crea attorno ad alcuni di essi una community che ne garantisce continui miglioramenti. I software liberi sono pertanto solitamente ben documentati e aggiornati frequentemente anche per una maggiore sicurezza.

Nel caso delle applicazioni Web, fortunatamente una gran parte delle tecnologie utilizzate per lo sviluppo e per i servizi di hosting è gratuita e disponibile come software libero.¹⁹ *JaLea* ad esempio, è stato sviluppato tramite la piattaforma LAMP²⁰ e utilizza software che non richiede costi di licenza.

2) Valutazione dell'acquisto di tema grafico o di funzionalità di organizzazione del layout

È sufficiente analizzare un sito Web recente di qualsiasi azienda importante, ad esempio la Sony, per riscontrare due elementi fondamentali: il tema grafico e la sua responsività.

Il tema grafico indica la struttura di base di un sito o un applicativo: la disposizione degli elementi, l'uso ragionato e uniforme dei colori, gli effetti animati quali slide in movimento o effetto di zoom nella selezione delle immagini. La responsività invece indica la funzionalità di disposizione automatica e modifica degli elementi di una pagina a seconda della grandezza della finestra o dello schermo in cui viene visualizzata. Inoltre, se una volta caricato il sito *sony.it*, si proverà a rimpicciolire la finestra, gli elementi si riposizioneranno e cambieranno di dimensione, mentre il menu in alto si trasformerà in un selettore che cliccato visualizzerà l'alberatura principale. Implementare elementi responsivi è una prassi ormai comune, in quanto gli utenti oggi utilizzano non solo il PC ma anche smartphone e tablet; è quindi necessario fare in modo che l'applicativo sia fruibile al meglio in dispositivi con schermi di dimensioni e proporzioni diverse.

Le immagini sottostanti tratte dal case study *JaLea* rappresentano la pagina principale del sistema e-learning visualizzata sia in uno schermo di computer sia in un tablet/smartphone verticale. Si noti come, in quest'ultimo, rispetto al primo i sottomenu principali diventa-

19 Wikipedia, «Software libero»: https://it.wikipedia.org/w/index.php?title=Software_libero.

20 Acronimo di Linux, Apache, MySQL e PHP. Si veda il paragrafo precedente per ulteriori dettagli.

no più grandi e si posizionano in verticale, mentre il menu scompare sostituito da un selettore posizionato alla destra della pagina Web.

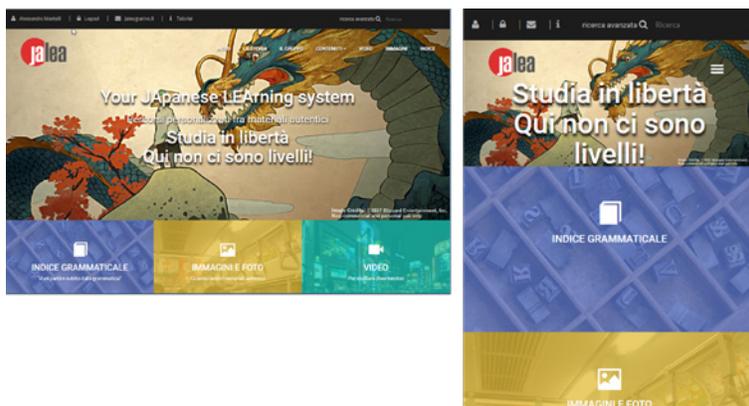


Figura 4.7 Disposizione degli elementi di *JaLea* sul desktop (immagine a sinistra) e su uno smartphone (immagine a destra)

Normalmente un'attività di design così attenta ai particolari e alla selezione dei colori, così come la programmazione del riarrangiamento del riposizionamento e trasformazione degli elementi nella pagina richiederebbe tempi considerevoli. In particolare, dovrebbe essere scritto molto codice non solo in HTML ma soprattutto in CSS,²¹ un linguaggio utilizzato per definire lo stile di ogni elemento HTML quale il colore, la posizione, la grandezza e la riallocazione in base alle dimensioni della finestra in cui la pagina viene eseguita.

Al fine di evitare la creazione da zero dell'ossatura della pagina con HTML e CSS si hanno due opzioni: utilizzare framework (gratuiti) per la creazione di elementi grafici standardizzati quali: *Bootstrap* o *Foundation*²² o acquistare un layout grafico preimpostato che contenga anche questi framework.

3) Verifica delle librerie e delle API disponibili per lo sviluppo

Le librerie sono collezioni di risorse prefabbricate che permettono di aggiungere nel proprio codice funzionalità di terze parti. Attraverso

²¹ Wikipedia, «CSS»: <https://it.wikipedia.org/w/index.php?title=CSS>.

²² Si vedano le rispettive pagine dei progetti: Bootstrap team (s.d.). *Bootstrap*. <https://getbootstrap.com>; Zurb Foundation (s.d.). *Foundation*. <https://get.foundation/>.

le librerie è possibile ad esempio, avere a disposizione meccanismi prefabbricati per esportare file in formato PDF o Excel, implementare funzionalità di gestione di dizionari di *kanji*, avere a disposizione interfacce dinamiche e responsive.

Oltre alle librerie interne all'applicativo, è possibile avere accesso a librerie delocalizzate. Queste ultime sono rese accessibili utilizzando le cosiddette API (*Application Programming Interface*).

Una API è una interfaccia che mette in contatto l'applicativo con delle librerie, anche esterne e molto spesso delocalizzate.²³

Google per esempio rilascia API per l'integrazione dei propri software principali: API per Calendar, Mail, YouTube, Google+, Tasks, nonché API per Machine Learning, servizi di traduzione o riconoscimento vocale.²⁴ Esistono API, inoltre, per tutti i Social media più conosciuti: Facebook, Instagram, Twitter, e API per servizi di conversione del testo in voce, o per l'interazione comunicativa uomo-macchina.²⁵

In *JaLea*, l'API di Google, YouTube, viene utilizzata nella sezione 'Video' per sincronizzare i sottotitoli con i video, sia in modalità di esecuzione libera che di ricerca di un punto specifico. Un'altra API, fornita liberamente dall'azienda giapponese Docomo,²⁶ permette la conversione automatica del testo in voce.

Quest'ultima API è stata utilizzata per replicare una funzionalità già presente in *BunpoHyDict* per ascoltare la lettura audio di tutti gli esempi in giapponese presenti nell'applicativo. Nel caso di *BunpoHyDict* era necessario che ogni singolo testo fosse letto e registrato da un parlante madrelingua giapponese, un metodo che permette sicuramente di avere un risultato di ottima qualità, ma che richiede tempi considerevoli e si ripercuote sulla manutenibilità. Bisogna infatti, non solo tenere conto della disponibilità dei lettori per il tempo effettivo di registrazione delle frasi, che si dilata in caso di errori del parlante, ma anche del tempo necessario per intervenire sul risultato del campionamento digitale, togliendo il fruscio e normalizzando il volume tramite un software apposito. In *JaLea* pertanto, sono stati mantenuti gli audio originali, quando disponibili, ma si è aggiunta anche la possibilità di convertire automaticamente un esempio scritto in parlato, qualora il file voce campionato non fosse presente. Creare da zero una funzionalità di conversione del testo in voce richiede

23 Le API possono essere disponibili su un server remoto e richiamabili solo attraverso una connessione Internet.

24 Per esempi di utilizzo di API di riconoscimento vocale in prototipi e programmi cf. Mantelli 2018 e Watanabe, Ōsaki 2018.

25 Per una lista di alcune API principali si veda ad esempio. Computer Science zone, *50 Most Useful APIs for Developers*: <https://www.computersciencezone.org/50-most-useful-apis-for-developers>.

26 Documentazione solo in giapponese. Docomo Developer support, «NTT Docomo»: <https://www.nttdocomo.co.jp/service/developer/>.

rebbe un gruppo di ricerca formato da programmatori, matematici, esperti di lingua giapponese e ingegneri del suono, generando costi e tempi di lavoro elevatissimi. La tecnologia di Docomo invece è sufficientemente matura da essere utilizzata in progetti di terze parti.

4.3.5 Affordance e signifier nella pratica del Web design

Come descritto nel capitolo 3, progettare oggetti complessi riflettendo su come realizzare affordance e signifier permette agli utilizzatori di interagire con tali oggetti in modo rilassato e spontaneo. Se l'affordance coadiuvato dal signifier è chiaro, l'utente percepisce immediatamente le modalità d'uso dell'oggetto. Per tale motivo questo una riflessione sulla progettazione visiva e funzionale degli elementi delle interfacce è fondamentale in ambito di user experience.

Guardando un oggetto, vediamo subito a cosa serve, se possiamo usarlo e come. Percezione e azione sono comprese in un unico atto. Questo è un cambiamento formidabile nella psicologia cognitiva, perché lo sviluppo del cognitivismo aveva spezzato il legame tra percezione e azione visti come fasi e processi cognitivi diversi. [...] Questo concetto permette di superare, nella gran parte dei casi, le fasi di elaborazione previste dal cognitivismo: siamo veloci, quando l'oggetto è usabile, diremmo 'è intuitivo'. (Bagnara 2017)

L'HTML fornisce di base una serie di oggetti con qualità intuitive non particolarmente efficaci, tuttavia l'abitudine dei primi utenti del Web ad interagire con tali oggetti ne ha reso esplicita la funzione a livello culturale. Ad esempio in una pagina Web un testo sottolineato e di colore blu, suggerisce la presenza di un hyperlink collegato sebbene l'affordance relativo non suggerisca immediatamente tale azione. La mappatura di collegamento tra l'hyperlink e la sua funzione (ovvero quella di aprire un'altra pagina) tramite il click del mouse è anche questo un processo più culturale che intuitivo. Analogamente i componenti radio e checkbox dell'HTML standard non posseggono alcuna proprietà per permettere di intuire che tali elementi siano cliccabili. L'utilizzo pertanto di librerie grafiche quali Bootstrap o di pacchetti di temi grafici strutturati non ha pertanto solo il fine espresso nel § 4.3.4 di permettere la manutenibilità dell'applicativo garantendo immediatamente funzionalità avanzate collaudate, ma anche di fornire al progettatore e allo sviluppatore elementi che posseggano un affordance più chiaro. Questo affordance inoltre viene progettato per essere adattato a più piattaforme: non solamente per il cursore del mouse ma anche per la pressione digitale tramite tablet e smartphone: il tasto pertanto diventa più grande, spesso con i bordi arrotondati a suggerirne la pressione; talvolta una leggera ombreggiatura lo

stacca dallo schermo per renderlo in maggiore evidenza. Il passaggio del mouse sopra il tasto spesso ne cambia il colore o ne evidenzia i bordi e il cursore stesso si trasforma in un'icona di selezione. Il signifier collegato al tasto può essere un termine che ne descrive la funzione o un'icona. In questo ultimo caso poiché il design delle icone rappresenta un'attività complessa e specializzata che richiede tempi notevoli ma tuttavia necessaria per esplicitare la funzione del signifier sul tasto, spesso vengono utilizzati pacchetti di icone professionali quali ad esempio *Fontawesome* (<https://fontawesome.com>).

Anche altri elementi disponibili nei form quali campi di input o selezioni a tendina, diventano più larghi per permetterne la pressione digitale. All'interno dei campi di input possono essere presenti dei placeholder (segnaposto) ovvero dei signifier testuali transitori che spariscono quando i campi vengono riempiti. Lo scopo del placeholder è quello di esplicitare il campo solamente quando serve senza utilizzare dei campi di testo permanenti (label). Le selezioni a tendina vengono dotate di funzionalità aggiuntive quali la possibilità di filtrare i campi presenti permettendo di identificare più velocemente i dati di cui si necessita. Relativamente ai tasti radio per le selezioni di due opzioni in contrasto quali ad esempio 'acceso/spento' 'attivato/disattivato' al posto degli elementi standard HTML vengono invece utilizzati dei controlli 'Toggle/Switch' (interruttori a levetta) in quanto l'affordance percepito esplicita maggiormente la funzione del controllo.



Figura 4.8 Esempio di elementi interattivi presenti in *JaLea*, che esplicitano il ruolo dell'affordance. Da sinistra a destra: tasti di selezione con testo, tasti di selezione con icona, radio switch, campo di input con placeholder e tendina di selezione con funzione di filtro incorporata

4.3.6 Utenti e interfacce

Dal punto di vista dell'interazione tra interfaccia e utente, le aree di backend e frontend di *JaLea*, sebbene caratterizzate da componenti grafici uniformi, sono state progettate con una alla base una visione differente. Questa scelta dipende dal fatto che come già spiegato nel § 4.3.3 gli utenti a cui sono destinate queste aree hanno differenti necessità.

L'area di backend, deve permettere all'amministratore e al collaboratore di poter identificare e modificare in breve tempo i record e pertanto la sua struttura tende a monopolizzare l'attenzione dell'utente attraverso l'uso di una grande tabella centrale filtrabile e ordinabile che contiene tutti i record e di form che occupano tutta la finestra organizzati in tab a seconda degli argomenti che appaiono al posto della tabella centrale in fase di aggiunta e modifica dei dati. Come già detto nel § 3.2, questo tipo di struttura prende il nome di *Sovereign posture* e ha la particolarità di presentare immediatamente all'utente l'area di lavoro su cui iniziare a lavorare. Al contrario l'area di frontend in uso allo studente di giapponese e descritta nel dettaglio nei §§ 4.5.1 e 4.5.2 prevede differenti percorsi di apprendimento ognuno dei quali è caratterizzato da interfacce di tipo differente. In questo caso lo scopo è fornire al discente le informazioni in modo progressivo al fine di evitare dei sovraccarichi di memoria. La navigazione dei contenuti multimediali ad esempio prevede un percorso in 2 step: il primo di selezione attraverso la visualizzazione di miniature di immagini o video, e il secondo attraverso la visualizzazione dell'elemento selezionato in posizione predominante e dei relativi elementi funzionali di contorno. Nel caso dei video, l'elemento principale è appunto il video stesso con i tasti di esecuzione sottostanti: stop e velocità, mentre gli elementi di contorno sono rappresentati dall'interfaccia di visualizzazione dei sottotitoli e selezione, degli elementi correlati e dell'area dedicata alla visualizzazione della traduzione italiana [fig. 4.21]. Nel caso degli elementi grammaticali, invece, l'interfaccia prevede una immagine esemplificativa dell'elemento grammaticale analizzato, segnalato da una freccia; una lista di dettaglio suddivisa in link a costruzioni grammaticali, applicazioni sintattiche nella lingua italiana e approfondimenti; i video e le immagini che lo contengono e quindi ne contestualizzano l'uso. Al fine di incentivare la capacità di identificazione degli elementi è stato fatto attento uso del colore per marcare i raggruppamenti dei link alle pagine di dettaglio. Si veda la spalla destra della finestra rappresentata in figura 4.13 (raggruppamenti dei link di dettaglio). La pagina di dettaglio di ogni elemento, che si raggiunge tramite i link della lista di dettaglio, prevede l'attivazione di una interfaccia apposita per la visualizzazione delle trascrizioni in *hiragana* e alfabeto latino, per il richiamo di un dizionario pop-up, e per la riproduzione dell'audio generato automaticamente dall'analisi del testo giapponese [fig. 4.16].

Un'altra strategia utilizzata per garantire una navigazione rilassata al discente è stata la creazione di scorciatoie intuitive di navigazione, quali ad esempio l'indice rapido (§ 4.5.1.4). Le scorciatoie di navigazione, infatti, permettono al discente di raggiungere più velocemente le informazioni di cui abbisogna seguendo un percorso alternativo una volta presa confidenza con il software.

4.4 Da *BunpoHyDict* a *JaLea*

JaLea è l'evoluzione con una nuova veste e un nuovo nome, del progetto *BunpoHyDict*, nato nel 2008, che nel tempo ha coinvolto oltre 72 collaboratori da tutto il mondo (Mariotti et al. 2017). Il motivo per cui non è stato possibile aggiornare ulteriormente *BunpoHyDict*, ove già era presente l'80% del contenuto attuale, è sicuramente l'obsoleta architettura tecnologica con cui era stato sviluppato.

La limitate competenze informatiche nel team di sviluppo iniziale, avevano portato infatti a utilizzare solo il linguaggio HTML, scelta che per il periodo storico (2008) era quasi obbligata e favorita dalla facilità con cui tale codice di programmazione poteva essere appreso e utilizzato.

Tuttavia, questa tecnologia non consentiva la crescita, l'utilizzo e la manutenibilità a livello economico e sviluppo di cui necessitava un progetto così ampio e ambizioso. Ogni pagina era gestita come file autonomo, copiando di volta in volta anche le informazioni riguardanti il layout principale. Pertanto, per modificare determinati elementi grafici o strutturali era necessario effettuare le modifiche in ciascuno dei file creati precedentemente alla modifica. Inoltre, il coinvolgimento di numerosi collaboratori, rendeva spesso difficile riconoscere i file più recenti da caricare online poiché era necessario nominarli con la medesima dicitura. Tale processo portava in qualche caso la perdita di importanti informazioni il cui recupero richiedeva tempi considerevoli. La scelta di utilizzare inoltre il formato MOV per i file video, supportato solo con l'installazione del plugin QuickTime aveva portato problemi di compatibilità con i browser più diffusi (Chrome, Internet Explorer e Mozilla Firefox) in quanto tale plug-in era basato sulla architettura NPAPI²⁷ che, considerata insicura, non è oggi più supportata dalle nuove versioni dei browser.

L'idea quindi di sviluppare il progetto *JaLea* nacque dalla necessità di poter creare un sistema e-learning mantenibile anche con risorse umane ed economiche relativamente limitate. Come membro del gruppo di ricerca del progetto (Mariotti, Mantelli, Lapis 2016), reso possibile grazie al supporto finanziario di 28.000 € di Mitsubi-

²⁷ Wikipedia, «NPAPI», <https://it.wikipedia.org/wiki/NPAPI>.

shi Corporation (2016), mi occupai della progettazione dell'esperienza utente e dello sviluppo informatico. I primi prototipi di *JaLea*, non erano ancora il risultato di una ricerca sistematica di una buona user experience, ma fu proprio questa ricerca, la relativa verifica della resa finale del design offerto al discente nell'area di frontend e il confronto con il team di ricerca a spingermi a intraprendere ricerche che considerassero fondamentale il ruolo del discente/utente nel processo di sviluppo. Questa attività di creazione, verifica e feedback descritta nel § 5.3 è alla base del modello di sviluppo ADDIE (§ 5.1.1).

JaLea inoltre, si sviluppò all'interno del Dipartimento di studi sull'Asia e sull'Africa Mediterranea (DSAAM) con l'obiettivo di dare risposta anche alle crescenti criticità di tipo didattico derivanti dai quasi 1.800 studenti di lingua giapponese (2015-16). Le classi di esercitazioni di lingua giapponese, nel 2015 vedevano infatti una proporzione docente/studente pari a 1/77, con picchi di 1/155 nelle classi di scrittura e composizione. Un rapporto così elevato richiedeva, dal punto di vista glottodidattico, soluzioni innovative e mirate, che solo un supporto di tipo informatico avrebbe potuto fornire. Il nuovo applicativo avrebbe quindi avuto la forma di un'applicazione Web, per permetterne l'utilizzo agli studenti indipendentemente dal luogo in cui si trovassero, e fornire uno strumento utile anche ai docenti, da utilizzare senza necessariamente installare un software nel computer a disposizione nell'aula assegnata per la lezione.

Dal punto di vista della tipologia di utenti a cui era destinato l'applicativo, si percepì come più urgente la calibrazione del prodotto sulle esigenze degli studenti del corso di laurea triennale, in quanto molto più numerosi rispetto agli studenti del corso di laurea magistrale, e quindi più bisognosi di strumenti dedicati all'apprendimento autonomo che supportasse una minore attenzione del docente ripartita fra troppi discenti.

A livello di progettazione dei contenuti e dei modelli di navigazione, il progetto *JaLea* ereditò da *BunpoHyDict* il concetto di fruizione del materiale grammaticale e multimediale attraverso l'uso estensivo di hyperlink, che permettono di collegare i testi di immagini e video alle schede grammaticali, e viceversa di passare dalle spiegazioni di grammatica alle immagini e video correlati. Si mantenne l'impostazione iniziale fondata sull'importanza di utilizzare materiali autentici in ambito glottodidattico rendendo pertanto possibile sperimentare l'uso della lingua giapponese in un contesto non manipolato. I materiali autentici infatti sono solitamente più attrattivi e interessanti e offrono quindi, dal punto di vista dello stimolo e dell'incentivazione del discente, la possibilità di una pianificazione di un'esperienza di apprendimento linguistico continuo (lifelong learning) (Mariotti 2011). Relativamente all'utilizzo dei materiali multimediali, studi sull'apprendimento dell'Inglese come seconda lingua hanno dimostrato fin dagli anni Novanta (Cronin, Cronin 1992; Liu 1995) co-

me questi siano particolarmente utili a contestualizzare i vocaboli e a favorirne l'apprendimento. Studi più recenti (Zhao, Susono 2018), inoltre, dimostrano analogamente la validità dell'utilizzo di materiali multimediali per l'apprendimento del giapponese.

I contenuti attuali di *JaLea*, in costante evoluzione, riprendono quelli originariamente presenti in *BunpoHyDict*, selezionati nel 2008-2010 in base alle unità 1-14 dello *Shokyū Nihongo*, ma l'inserimento di materiali autentici multimediali in questo tipo di applicazione comporta la corrispondente creazione di nuove voci grammaticali legate a ciascun materiale autentico fornito e quindi selezionate non in modo lineare. Il processo di apprendimento può attivarsi sia dalla tradizionale grammatica affrontata in classe e approfondita poi dal discente tramite *JaLea*, che viceversa, a partire dalla scelta del materiale autentico che il discente ritiene più interessante, stimolando in quest'ultimo percorso l'emisfero cerebrale legato al piacere e quindi alla memoria a lungo termine. Alcuni dei parametri per la selezione dei materiali riguardano lo *span* di attenzione audio-visiva²⁸ e la contemporaneità (da qui la necessità di un continuo aggiornamento dei materiali e rispettivi contenuti) ed eventuale disponibilità dei diritti. Per quanto riguarda gli esempi a supporto delle spiegazioni grammaticali, in *JaLea* oltre ad esempi originali, così come avveniva in *BunpoHyDict*, si è aggiunta una funzione per citare la fonte di esempi eventualmente tratti da libri di testo.

Come indicato anche nel capitolo 2, si è compreso che la navigazione libera dei contenuti, sebbene di estrema novità dal punto di vista dell'esperienza dell'utente in logica costruttivista, avrebbe potuto mettere in difficoltà l'utente abituato a percorsi sequenziali, che si rifanno in qualche modo ai libri testo tradizionali. Era quindi necessario stimolare il discente su vari piani. Non solo cioè era necessario fornire un valido metodo di navigazione di contenuti selezionati, valutati in base all'analisi dei bisogni dei discenti di giapponese di Ca' Foscari, ma era altresì necessario considerare l'usabilità dell'interfaccia, l'aspetto grafico e l'organizzazione spaziale degli elementi attraverso un design del layout della pagina che non provocasse smarrimento.

Il team quindi ha innanzitutto valutato i requisiti di sviluppo e di descrizione delle funzionalità, per studiare come progettare l'interfaccia dal punto di vista della struttura e dei meccanismi di utilizzo, nonché per definire un impianto tematico e grafico coerente, adattabile ai vari dispositivi. Tale ricerca doveva considerare però anche le limitazioni del budget, e la necessità di manutenibilità del progetto.²⁹

28 Secondo uno studio di HubSpot, la lunghezza massima ideale di un video per catalizzare l'attenzione degli studenti è di due minuti (Chi 2018).

29 Si veda il modello di Keeley 2013 del § 3.4.

La ricerca si è estesa quindi da una analisi dei contenuti e delle modalità di fruizione nell'e-learning, a un'analisi più completa e multilivello, che considera il sistema e-learning all'interno dell'ecosistema uomo-macchina, considerando quindi l'interfaccia, i parametri di usabilità e l'impianto grafico/strutturale, orientata pertanto in un'ottica di experience design.

4.5 *JaLea*, Your Japanese Learning System!

JaLea è un'applicazione web³⁰ per l'acquisizione della lingua giapponese da parte di italofoeni, indirizzata a superare i limiti legati a materiali didattici convenzionali, quali libri di grammatica o manuali di lingua giapponese.³¹

Sebbene *JaLea* attualmente sia indirizzato solo a un pubblico italofono, è stato previsto un backend che consenta in futuro di crearne versioni per discenti di lingue differenti, anche non europee. Il sistema intende superare alcuni dei limiti legati ai materiali convenzionali:

- metodi costrittivi, noiosi e dispendiosi a livello di tempo, a causa di metodi di insegnamento tradizionali insegnante-discente (top down) o, al contrario, materiali amatoriali, non accurati rivolti a discenti non inseriti nei sistemi di istruzione universitaria;
- frammentazione dei materiali di riferimento;
- materiali artificiali, creati al solo fine didattico, che non consentono di relazionarsi con una lingua straniera contestualizzata in un ambiente reale.

JaLea è stato pensato per essere usato sia per l'apprendimento in autonomia, sia come grammatica di riferimento in supporto alle lezioni del docente. Lo scopo dell'applicativo è quello di collegare l'apprendimento della lingua a un'attività piacevole (Balboni 2012), rispettando quindi i tempi individuali di studio e l'autonomia del discente, nonché di permettere al docente di svolgere un nuovo e fondamentale ruolo come educatore di cittadini globali attraverso il dialogo, come suggerisce la pedagogia critica (Freire 1970; Hosokawa et al. 2016).

Dal punto di vista dell'utilizzo dell'applicazione Web, la struttura di *JaLea* può essere divisa in due parti principali: l'area studente (frontend), che per funzionalità richiama il concetto definito nel processo di experience design come area di front-office, e l'area ammi-

³⁰ <https://jalea.unive.it/jalea/it/home>.

³¹ Il presente capitolo è frutto dei feedback ottenuti dalle presentazioni ai convegni: Head'17 Conference on Higher Education Advances, Valencia 2017 e The 15th International Conference of the European Association for Japanese Studies EAJE 2017, Lisbona.

nistratore (backend), definita nel processo di experience design come area di back-office (§ 3.5).

4.5.1 *JaLea*: frontend e struttura

L'interfaccia di *JaLea* è caratterizzata da due elementi sempre presenti: 1) l'area con sfondo nero in cima alla pagina contiene i tasti di login, logout, il campo di ricerca semplice e il tasto di ricerca avanzata e 2) il menu sottostante l'area nera, ordinato a destra dello schermo. In questa area è presente anche il tasto [Tutorial], azionando il quale è possibile visualizzare una finestra di pop-up con un video introduttivo all'utilizzo dell'applicativo.

Nella pagina principale sono presenti tre grandi tasti di colore differente che ne indicano i contenuti: [Indice grammaticale], [Immagini e foto], [Video].

L'accesso a *JaLea*, è attualmente, permesso solo attraverso l'uso delle credenziali dell'Università Ca' Foscari. Cliccando il tasto [Login] lo studente viene ridiretto alla pagina di inserimento credenziali di ateneo e solo una volta completato l'inserimento delle credenziali corrette, egli viene ridiretto alla pagina principale dell'applicativo. La registrazione consente di utilizzare anche materiali protetti da copyright e procedere allo sviluppo di funzionalità che ne precludano la condivisione indifferenziata.

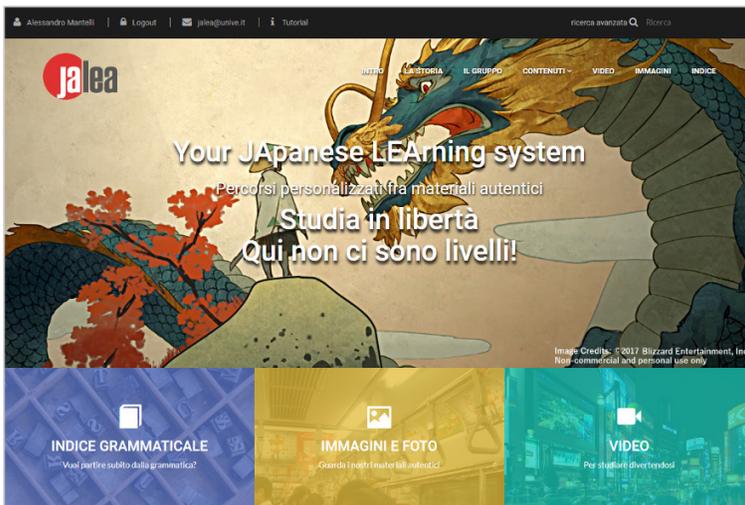


Figura 4.9 *JaLea*, pagina iniziale

Inoltre, come indicato successivamente nel capitolo 5, l'utilizzo delle credenziali univoche di ateneo attraverso il sistema Shibboleth per accedere a *JaLea* permette di tracciare in modo univoco gli studenti, e di utilizzare i dati ai fini statistici.

4.5.1.1 Il menu principale

Il Menu principale è composto da sette voci: 'Introduzione', 'La storia', 'Il gruppo', 'Contenuti', 'Video', 'Immagini', 'Indice'.

Introduzione

Questa sezione offre al principiante assoluto, informazioni essenziali sulla lingua giapponese. La sezione è organizzata in otto tab.

Nel primo tab, le caratteristiche principali della lingua giapponese sono sommarizzate in sette punti.

Grammatica

Figura 4.10 *JaLea*. Istantanea tab, sezione Introduzione

Scrittura 1, scrittura 2

Vengono descritti in modo schematico i sistemi di scrittura in uso alla lingua giapponese moderna. In particolare, l'uso dei *kanji*, dell'*hiragana* e del *katakana*, nonché come nel testo giapponese convivano caratteri e orientamenti di scrittura molteplici.

Il gruppo

Attraverso questa voce di menu si possono ottenere informazioni sul gruppo di ricerca del progetto *JaLea*, formato da Marcella Mariotti (*principal investigator*), Alessandro Mantelli (*system developer*) e Giovanni Lapis (*content manager*).

Contenuti

Cliccando sulla voce di menu [Contenuti], presente in alto a destra della pagina, lo studente può visualizzare l'indice dei contenuti relativi alla grammatica giapponese, indicizzati per l'iniziale in alfabeto latino della voce desiderata. Questa è una delle componenti fondamentali di *JaLea* pertanto è possibile aprire il menu contenuti anche attraverso un tasto sempre presente in alto a sinistra nella pagina e denominato 'Indice'.

Indice

L'indice completo di tutti i contenuti può essere visualizzato cliccando la voce di menu 'Indice' o il riquadro viola 'indice grammaticale' della pagina principale

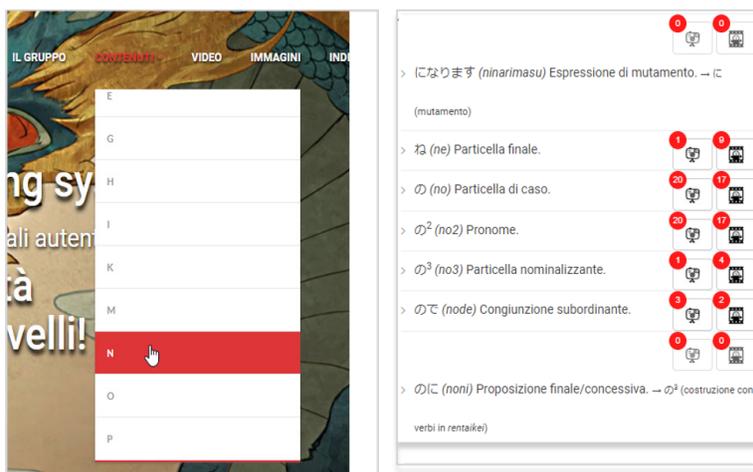


Figura 4.12 Indice di *JaLea*. A sinistra il sottomenu che porta alla lista delle voci di destra

Come si nota dalla immagine a destra, per ogni voce dell'indice sono presenti due icone con un pallino rosso ed un numero all'interno. Questo numero indica quante volte la voce è presente nei contenuti multimediali relativi alle immagini (piccola icona a sinistra) e ai video (piccola icona a destra).

I contenuti sono organizzati a due livelli principali: 'descrizione' e 'dettaglio'.

4.5.1.2 Area 'Descrizione'

A questo livello vengono visualizzate le informazioni generiche e comuni di un determinato elemento grammaticale. Ogni schermata presenta una descrizione, un'immagine relativa a un contenuto autentico (pubblicità nel treno o nella stazione, un cartello stradale, una insegna) e una serie di link di dettaglio sui casi d'uso dell'elemento grammaticale in analisi.

Sull'immagine centrale è posizionata una freccia che indica un elemento grammaticale relativo alla scheda presentata. Cliccandola, si apre la scheda di dettaglio relativa all'elemento grammaticale nel contesto dell'immagine.

I link di dettaglio, presenti nel lato destro sono divisi in più gruppi a seconda della forma grammaticale presa in esame. I gruppi più utilizzati sono: 'Complementi', 'Costruzione', 'Subordinate' e 'Approfondimenti'.

Sotto la sezione dei link di dettaglio, è presente l'elenco degli elementi grammaticali correlati a quello preso in esame. Questa struttura a nodi interconnessi e non necessariamente sequenziale, è l'elemento più innovativo e fondamentale del sistema e-learning *JaLea*, che riprende ottimamente le teorie della motivazione all'apprendimento di Siemens (2005) dove il discente segue interessi e percorsi individuali tramite connessioni e mappe mentali personali, e consente di considerare *JaLea* un sistema e-learning altamente inclusivo, che cioè viene incontro alle necessità di utenti con processi di apprendimento non necessariamente convenzionali.

Di seguito, sempre nel lato destro dello schermo, sono presenti le miniature dei video e delle immagini che contengono tale elemento. Il numero nell'angolo in alto a destra di ogni miniatura indica la frequenza con cui l'elemento grammaticale è presente nel contenuto multimediale. Cliccando la miniatura, lo studente viene diretto alla sezione 'Immagini' o 'Video' corrispondente.

4 • Progettare un e-learning per il giapponese. Il case study *JaLea*

The screenshot shows the JaLea website interface. At the top, there is a navigation bar with links like 'INIZIO', 'LA STORIA', 'IL GRUPPO', 'CONTENUTI', 'VIDEO', 'IMMAGINE', and 'INDICE'. The main content area is titled 'Descrizione' and features a large image of a Japanese book cover with the title '車内でのマナーとルール' (Manners and Rules in the Car). Below the image, there are sections for 'Significato', 'Denominazione grammaticale', and 'Traduzione approssimativa'. On the right side, there is a sidebar with various categories of grammatical points, each with a colored header: 'Complementi' (pink), 'Coniugazione' (pink), 'Subordinate' (pink), 'Approfondimenti' (green), and 'Elementi Correlati' (blue). At the bottom of the sidebar, there is a 'Foto' section with a grid of small images. Yellow callouts with arrows point to specific elements on the page, such as the 'indice' button, the 'ni' icon, the book image, and various sidebar categories.

Figura 4.13 *JaLea*. Pagina di Descrizione di una voce grammaticale

La presentazione dei contenuti avviene utilizzando colori differenti per ciascuna categoria di link alle pagine 'Dettaglio' (Costruzione, Complementi, Subordinate, Utilizzi in vari contesti, Approfondimenti) e per le varie sezioni, in modo da aiutare lo studente nel processo di distinzione secondo il principio dei segnali di Mayer.³² Secondo questo principio infatti, un migliore trasferimento delle informazioni avviene quando la narrazione è corredata da elementi identificativi che ne esemplificano la funzione quali icone, colori e elementi grafici.

³² Per la lista dei principi di Mayer si veda il § 2.4.

4.5.1.3 Area ‘Dettaglio’

I link di dettaglio nella pagina di descrizione collegano a pagine in cui vengono descritti i vari usi nella lingua italiana di un determinato elemento grammaticale giapponese. La scelta di fare riferimento alle categorie grammaticali italiane consente infatti di facilitare il processo di apprendimento in quanto come suggerito dall'ipotesi dell'input di Krashen (1985), le nuove conoscenze sono più assimilabili se costruite su conoscenze pregresse.

I contenuti nell'area di dettaglio possono variare a seconda dell'elemento grammaticale analizzato, ma tendenzialmente mostrano la seguente struttura:

- titolo dell'elemento grammaticale preso in esame;
- informazioni essenziali sull'uso. Esempio: N (luogo) ㇿ ni V (moto);
- tabella con un esempio analizzato nel dettaglio;
- spiegazione;
- esempi.

Per ogni esempio è possibile ascoltare l'audio corrispondente cliccando l'icona a forma di cassa musicale a sinistra di questo. Ci sono due modalità: voce reale e voce sintetizzata.

La modalità voce reale, identificata dall'icona , permette di ascoltare la lettura del testo in giapponese svolta da un lettore madrelingua. Per tutti i casi in cui questo non è possibile per mancanza del file, è possibile ascoltare la lettura del testo attraverso un processo di sintesi vocale; il testo in pratica viene automaticamente convertito in voce. Questa ultima funzionalità viene indicata dall'icona .

Anche in questo caso, l'uso attento dei colori per identificare le zone di interesse e delle icone per identificare funzionalità multimediali, permettono allo studente di avere le informazioni essenziali per la navigazione, ma senza riempire in modo eccessivo la pagina. L'utilizzo della tecnica di navigazione frammentaria (*breadcrumb*), permette allo studente di tenere traccia della propria posizione nella attività di navigazione.

The screenshot shows the JaLea interface for the particle 'ni'. At the top, there is a navigation bar with 'JaLea' logo and menu items: INTRO, LA STORIA, IL GRUPPO, CONTENUTI, VIDEO, IMMAGINI, and INDICE. Below the navigation bar is a breadcrumb trail: 'Complementi / Complemento di stato in luogo > [ni]'. The main content area is titled 'N (luogo) に V (esistenza)'. It features a table with columns for the particle 'ni' and its usage. The table includes examples like 'ジジさん' (Gigi), 'は' (wa), '大学' (università), and 'います' (esistere). A callout 'Icone di selezione audio' points to the 'ni' row. Below the table, there are sections for 'Esempi' (Examples) and 'Approfondimenti' (In-depth notes). The 'Approfondimenti' section contains a note about the particle 'ni' being used for location and direction. At the bottom, there is a callout 'Interfaccia a fondo pagina' pointing to a footer area with 'JLPT' and 'ABC' buttons.

N	ジジさん ジジ san	Gigi
P	は wa	(tema)
N	大学 だいがく	università
P	に ni	in
V	います imasu	esistere

Figura 4.14 *JaLea*. Esempio pagina di dettaglio

4.5.1.4 Funzionalità generali

Di seguito la descrizione delle funzionalità comuni a tutte le pagine dell'applicativo.

Indice rapido

Per permettere allo studente di accedere comodamente e più rapidamente alle funzionalità dell'indice anche da smartphone o tablet, è stato introdotto un tasto rapido di indice nella parte in alto a sinistra della pagina. Infatti, quando la finestra del device in uso è stretta orizzontalmente, il menu principale si trasforma in un'icona. Per-

tanto, per selezionare una voce grammaticale sono necessarie ogni volta tre azioni: 1) pressione dell'icona di menu a tre linee orizzontali; 2) selezione della voce 'Contenuti', e 3) selezione della sottovoce desiderata. Per agevolare la ricerca è stata introdotta una funzione, utilizzabile nell'applicativo sia da PC che da dispositivo mobile, che permette di visualizzare in sovrapposizione tasti sufficientemente grandi da poter essere premuti facilmente, anche in dispositivi relativamente piccoli, con le lettere alfabetiche rappresentanti i filtri dei contenuti grammaticali.

Una volta cliccato il tasto rapido [Indice], il menu con le iniziali rimane sempre aperto in sovrapposizione, e sempre visibile anche in caso di scorrimento verticale dei contenuti.

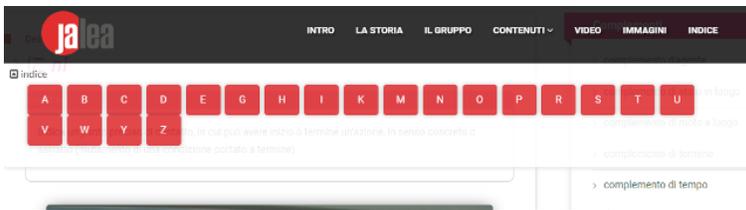


Figura 4.15 *JaLea*. Indice rapido

Dal punto di vista delle strategie di user experience, la possibilità di utilizzare questo menu permette allo studente di essere più rapido nella navigazione dei contenuti, evitando eventuali sovraccarichi cognitivi che porterebbero alla rottura del processo di estensione fisica del discente all'artefatto digitale (embodiment).

Interfaccia a fondo pagina

A fondo pagina è presente un'altra interfaccia in sovrapposizione, con tre tasti sul lato sinistro e un tasto sul lato destro. Anche questa interfaccia è sempre presente, anche in caso di scorrimento verticale



Figura 4.16 *JaLea*. Dettaglio dell'interfaccia a fondo pagina

I tasti [*furigana*] e [*rōmaji*] permettono di attivare e disattivare automaticamente la trascrizione in *furigana* e caratteri latini di ogni testo giapponese presente nella pagina. Le impostazioni vengono me-

morizzate nel browser dello studente, in modo da permettergli di visualizzare tutti i testi successivi nel formato più adatto alle proprie competenze linguistiche.

Attraverso il tasto [dizionario] è possibile attivare la funzionalità omonima. Con questa funzionalità attiva, posizionando il mouse sul testo giapponese, apparirà una finestra in pop-up che mostrerà la relativa traduzione in italiano (se presente) o in inglese, nonché l'animazione dei tratti del termine selezionato. Ad esempio, posizionando il mouse sul termine 図書館 (*toshokan*), apparirà la finestra di pop-up come da figura 4.17. Nel caso dei verbi, il pop-up indicherà anche a quale forma verbale appartengono e quale è la forma base. Oltre alla traduzione in italiano del termine, un'animazione mostrerà il corretto ordine dei tratti per ogni carattere in esame.



Figura 4.17 *JaLea*. Funzionalità di dizionario con tratto automatico dei *kanji* e indicazione della forma verbale

Infine, il tasto sulla destra permette di ritornare in cima alla pagina, funzionalità molto utile in caso di contenuti numerosi.

4.5.1.5 Immagini e video

Cliccando sulle voci del menu principale 'Immagini' e 'Video', l'utente può accedere alle relative sezioni dell'applicativo. In entrambe le sezioni sono presenti immagini e video di materiali autentici che presentano reali situazioni in Giappone, senza nessuna manipolazione dei contenuti per adattarli allo scopo didattico.

Tutti i materiali sono autentici e sono stati selezionati attentamente, ipotizzando quali elementi della vita giapponese possano essere particolarmente interessanti per lo studente, al fine di suscitare un

sentimento di piacere nell'attività di esplorazione dei contenuti (Mariotti 2015).

Cliccando [Immagini], apparirà una lista di miniature con il titolo e una porzione dell'immagine.



Figura 4.18 *JaLea*. Pagina di selezione immagini

Cliccando una delle miniature, si aprirà una pagina che contiene l'immagine nella sua interezza e la trascrizione del testo al suo interno. Cliccando [Visualizza Traduzione] verrà visualizzata la traduzione in italiano del testo. È stato volutamente scelto di non visualizzare subito la traduzione, affinché lo studente, se lo desidera, possa cimentarsi personalmente nella traduzione del testo giapponese prima di verificarne la correttezza. Nel lato destro della pagina è presente la lista di tutti gli elementi correlati del testo che appare nell'immagine.

The screenshot shows the JaLea website interface. At the top, there is a navigation menu with links for 'INTRO', 'LA STORIA', 'IL GRUPPO', 'CONTENUTI', 'VIDEO', 'IMMAGINE', and 'INDICE'. Below the menu is a search bar and a header with the JaLea logo. The main content area is titled 'Immagine' and displays a photograph of a sign. The sign has a red prohibition symbol and the text 'ゴミは忘れずにお持ち帰りください' (Please don't forget to take your trash home). Below the image, there is a 'Testo' section with the text 'すべてのお客様が快適にご利用いただけるよう、ゴミは忘れずにお持ち帰りください。' and a 'Visualizza Traduzione' section with the translation 'Non dimenticate di portare via i vostri rifiuti per favore, in modo che tutti gli ospiti possano usufruire confortevolmente [dei nostri servizi].'. On the right side, there is a sidebar titled 'Elementi Correlati' (Related Elements) with a list of grammar points such as 'Particella di caso', 'Particella di cortesia', 'Particella di caso (soggetto)', 'Particella di caso e coniugazione avverbale dell'ausiliare pseudo-verbale "shikusu"', 'Modo di ribaltone', 'Potenziale', 'Espressori o forme verbali per esprimere abilità o possibilità di eseguire un'azione', 'Particella tematizzante', 'Diffesso negativo del verbo', 'Verbi a cinque vocali, verbi del gruppo 1, verbi consonantici', and 'Verbo di dono'.

Figura 4.19 *JaLea*. Pagina di dettaglio di un'immagine

Come si può vedere dalla figura 4.19, il testo dell'immagine, presenta molti termini in colore viola. Questo colore identifica gli hyperlink che se cliccati, aprono la pagina con la spiegazione grammaticale corrispondente.

Lo studente pertanto può continuare ad approfondire il proprio studio attraverso un processo di navigazione continua, che lo porta ad apprendere successivamente i vari dettagli sulla grammatica secondo i propri bisogni.

Anche in questa sezione è possibile utilizzare le funzionalità del menu di fondo pagina e relativo dizionario.

La fruizione dei video funziona in modo analogo a quello delle immagini. Cliccando [Video] dal menu principale, appare una selezione dei video presenti nel database di *JaLea*.

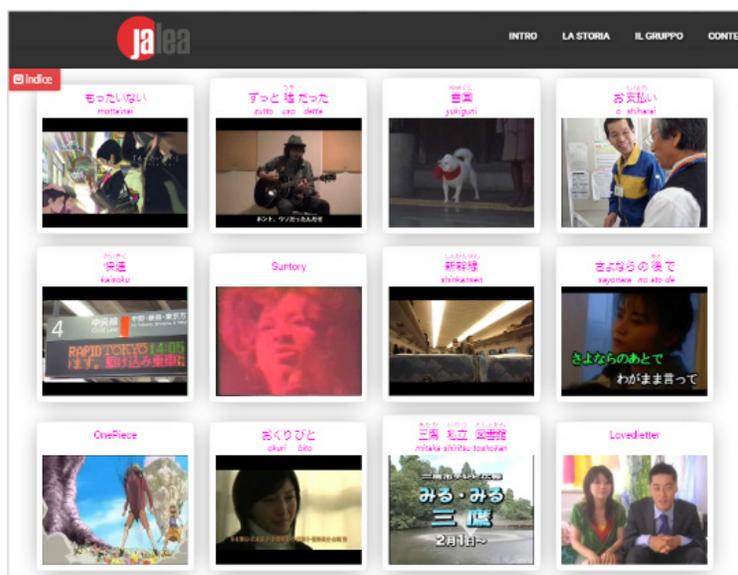


Figura 4.20 *JaLea*. Pagina di selezione video

Cliccando una miniatura, apparirà la pagina del video selezionato. La pagina propone a sinistra il video e a destra i relativi sottotitoli. Nel caso di smartphone e tablet lo schermo verticale dispone i sottotitoli sotto il riquadro del video.

Con l'esecuzione del video, il sottotitolo corrispondente viene automaticamente evidenziato in grassetto. È inoltre possibile cliccare sul minutaggio (timeline) di ciascun sottotitolo per avviare il video dal momento desiderato. Cliccando il tasto con l'icona del cronografo, a destra del tasto [play] si può rallentare la velocità di riproduzione, consentendo al discente di esercitarsi nella comprensione orale o nella trascrizione di quanto sente. Anche in questo caso, come da figura 4.21, alcuni elementi del testo sono hyperlink, che una volta cliccati collegano il discente alla pagina di descrizione dell'elemento grammaticale corrispondente. Sulla parte destra della pagina inoltre vengono elencati gli elementi grammaticali correlati, ovvero presenti nel testo del video o dell'immagine; ogni tasto, se cliccato, porta alla pagina di descrizione del contenuto grammaticale desiderato. Analogamente alla sezione 'Immagini', anche nella sezione video è possibile utilizzare le funzionalità automatiche di trascrizione del testo giapponese e del dizionario, tipiche di *JaLea*.

The screenshot shows the JaLea website interface. At the top, there is a navigation bar with the JaLea logo and menu items: INTRO, LA STORIA, IL GRUPPO, CONTENUTI, VIDEO, IMMAGINI, and INDICE. Below the navigation bar, there is a red 'Indice' button. The main content area is titled 'Video' and features a video player with the title 'お支払い o shiharai'. The video player shows a scene with two people at a counter. To the right of the video player, there is a list of numbered items (01, 18, 25, 42, 55, 185) with Japanese text and Romanized examples. Below the video player, there is a 'Visualizza Traduzione' button and a 'Fonti e commenti' section. On the right side of the page, there is a 'Elementi Correlati' sidebar with a list of related grammatical elements, including indicators of distance, auxiliary pseudo-verbals, adverbs, terminations, and verb variations.

Figura 4.21 *JaLea*. Dettaglio di una pagina video

4.5.2 *JaLea*: proposte di utilizzo

Uno fra i tanti aspetti innovativi e tuttora originali, di *BunpoHyDict* e della sua evoluzione contemporanea *JaLea*, è sicuramente la possibilità di attingere a informazioni su una determinata struttura grammaticale interna a un testo. Estensioni per browser quali *Rikaichan*, *Rikaikun* o *Yomichan* permettono solo di ottenere la traduzione di un termine ma non di conoscerne la funzione grammaticale. *JaLea* invece consente a qualsiasi persona, principiante o meno, che desideri ad esempio leggere un articolo di giornale online, di affiancare ai dizionari pop-up, uno strumento efficace per comprenderne non solo il vocabolario ma anche la sintassi e la grammatica.

Utilizzando *JaLea* cioè, è possibile selezionare attraverso l'indice una particella o un costrutto, accedere alla pagina di descrizione del costrutto, e da lì alla pagina dei dettagli, individuando così tutte le possibili funzioni grammaticali dell'elemento selezionato. Nella scheda di 'Descrizione' della grammatica, il discente ritroverà una semplice ed immediata spiegazione sulla voce consultata, l'elenco dei casi d'uso relativi, e le miniature di immagini e video la riportano, mentre nella pagina 'Dettaglio' accederà ai singoli esempi di utilizzo. Per maggiori informazioni potrà quindi scegliere di accedere al materiale multimediale, o, attraverso gli hyperlink, ad altre voci grammaticali correlate.

Un percorso alternativo al primo, avrà invece inizio dal materiale multimediale, video e immagini, da cui risalire alle pagine di 'Dettaglio' grammaticale tramite gli hyperlink inseriti in ciascuna trascrizione.

Un terzo metodo per utilizzare *JaLea*, proprio solo di questa evoluzione e non del 'genitore' *BunpoHyDict*, a causa delle sue restrizioni tecnologiche, è la funzionalità 'Ricerca', declinata secondo molteplici filtri attivabili. In testa alla pagina Web dell'applicazione si può selezionare la voce 'Ricerca' o cliccare l'icona a forma di lente [ricerca avanzata], tramite cui è possibile inserire nel una parola chiave e ricercarne l'occorrenza in tutti i contenuti presenti in *JaLea*.

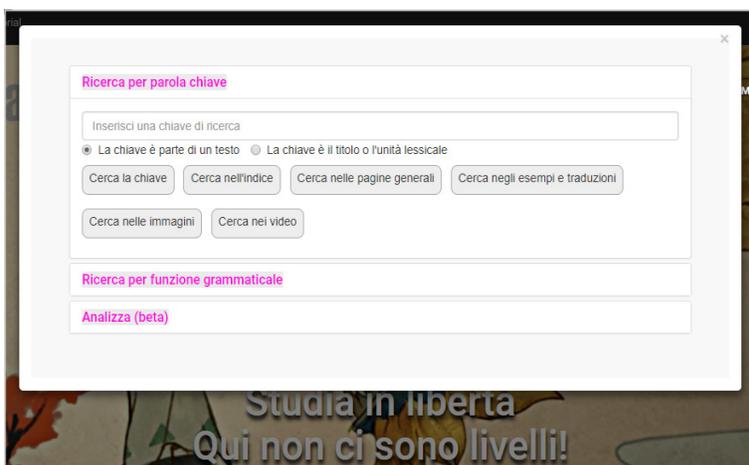


Figura 4.22 *JaLea*. Funzionalità di ricerca

Selezionando [ricerca per parola chiave] e inserendo come parola chiave *に ni*, ad esempio, sarà possibile ricercarne tutte le occorrenze nei testi di esempi, nelle immagini e nei video.

The screenshot displays the JaLea search interface. At the top, there's a navigation bar with 'INTRO', 'LA STORIA', 'IL GRUPPO', 'CONTENUTI', 'VIDEO', and 'IMMAGINE'. Below it, a search bar shows 'Risultati della ricerca per: に' and 'Ho trovato 203 risultati cercando ovunque'. The main content area is divided into several sections:

- Sezione descrizione:** A yellow callout points to the 'Descrizione' section for the character 'に', which states it is a particle and conjugation of the pseudo-verbal auxiliary 'da/desu'.
- Sezione dettaglio:** A yellow callout points to the 'Costruzione' section for 'もう・Averbio', which includes an example sentence: '[...]in, l'ho già visto. 欠席についてもう担当の先生に連絡済み。 Ho già avvisato della mia assenza il pr[...]']
- Estratto del testo corrispondente la ricerca:** A yellow callout points to the 'Costruzione' section for 'まだ・Averbio', which includes an example sentence: '[...]ancora mangiato? 欠席についてまだ担当の先生に連絡していません。 Non ho ancora avvisato della mia as[...]']

Below these, there's another 'Costruzione' section for '前に・Averbio' with an example sentence: '[...]Nの前にリュック(zain)は ((tema)) 自分(sé stesso) の (di) 前に (davanti) 持つ/持ちます (portare). [...]'.

Figura 4.23 *JaLea*. Risultato di una ricerca generica. Vengono cercate tutte le occorrenze della chiave di ricerca に *ni* anche dentro gli esempi

Utilizzando il filtro di ricerca [la chiave è parte di un testo], verrà ricercato il termine in qualsiasi scheda di *JaLea*. Come in figura 4.23, ad esempio, sebbene sia stata cercata la particella に *ni*, i risultati la riportano sia dalla pagina 'Descrizione' corrispondente, sia dai vari esempi delle pagine di 'Dettaglio' e 'Costruzione'. Il testo nel riquadro grigio rappresenta un estratto in cui la chiave ricercata appare, in modo da permettere allo studente di capire se il risultato è conforme alle sue aspettative. Come descritto in figura 4.24, è possibile utilizzare differenti filtri, al fine di svolgere ricerche particolarmente affinate. In particolare, è anche possibile effettuare ricerche in base alla funzione grammaticale desiderata. Cliccando questa opzione, apparirà una lista di tag grammaticali. È possibile selezionarne molteplici contemporaneamente per effettuare ricerche combinate.

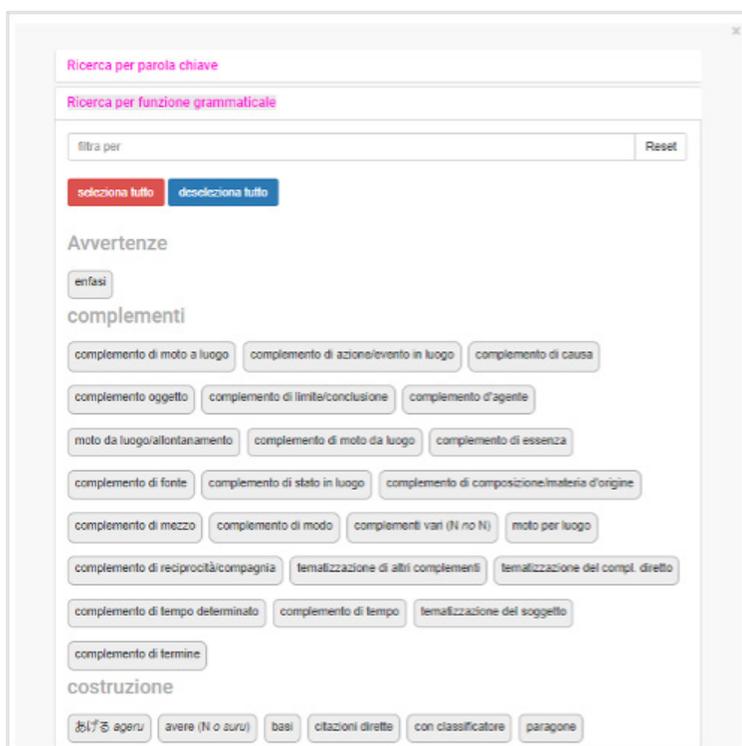


Figura 4.24 *JaLea*. Ricerca per funzione grammaticale

4.5.3 *JaLea*: backend

L'area di backend è l'area di amministrazione attraverso la quale il content manager può inserire i contenuti. Vi si accede attraverso un indirizzo Web differente inserendo specifiche credenziali che possono essere create attraverso una funzionalità presente nello stesso backend.



Figura 4.25 *JaLea*. Pagina di richiesta credenziali per l'accesso a *JaLea* o altri progetti e-learning gestiti dal gruppo di ricerca

Attraverso le varie funzionalità del backend è possibile principalmente:

1. aggiungere, modificare, eliminare unità lessicali;
2. aggiungere, modificare, eliminare immagini;
3. aggiungere, modificare, eliminare video;
4. aggiungere, modificare, eliminare pagine di 'Dettaglio'.

Ogni menù dell'area di backend richiama una lista che permette una visione d'insieme dei contenuti. Cliccando una riga della lista, si apre il dettaglio del record relativo. Ogni lista inoltre è filtrabile per parola chiave, e ogni colonna è ordinabile alfanumericamente.

Si veda ad esempio, la figura 4.26 che riguarda la sezione relativa all'inserimento e la modifica delle pagine 'Descrizione'.

id	Pubblica	Pubblico giorno	Unità lessicale	Indice	Tipo	Tags	Index tags	Denominazione grammaticale	m_attr1	m_attr2	m_attr3	tag
1	si	2016-08-24 06:27:00	IC	ni	Descrizione		IC	Esempio di caso c con la grammatica aritmetica dell'ausiliare pseudo-verbale 'daru'.				si,ni
2	si	2016-08-25 12:09:00	IC	ni	Dati	complemento d'agente	IC	Agente				si,ni
80	si	2016-10-18 00:42:00	IC	ni	Itati	complemento di stato in luogo						
81	si	2016-10-18 02:53:00	IC	ni	Dati	mutamento						
85	si	2016-10-18 05:53:00	IC	ni	Itati	complemento di modo						
82	si	2016-10-18 05:58:00	IC	ni	Itati	complemento di modo a luogo						

Figura 4.26 *JaLea*, area di backend. Lista delle unità lessicali

Cliccando il record con ID: 1, si aprirà un'area di dettaglio che permetterà l'inserimento dei contenuti nei molteplici campi relativi.

Figura 4.27 *JaLea*, area di backend. Interfaccia per l'inserimento dei contenuti

L'area di dettaglio è divisa in 5 sotto-schede, ognuna dei quali contiene i corrispettivi campi dedicati alla modifica di informazioni:

Il tab 'Principale' permette di inserire le seguenti informazioni:

- pubblicazione e data di pubblicazione della scheda;
- tipologia di scheda: Descrizione o Dettaglio;
- voce da visualizzare: in questo caso ㇷゝ;
- denominazione grammaticale;
- traduzione approssimativa;
- significato;
- immagine di descrizione.

Nel tab 'Correlati' è possibile selezionare quali tra le schede inserite sono collegate a questa. Nel tab 'Pointer', è possibile posizionare una freccia sull'immagine di descrizione per evidenziare all'interno del materiale autentico utilizzato la parte di esempio della voce che si sta inserendo, collegandovi una scheda di dettaglio che ne semplifica l'uso specifico all'interno della frase riportata.

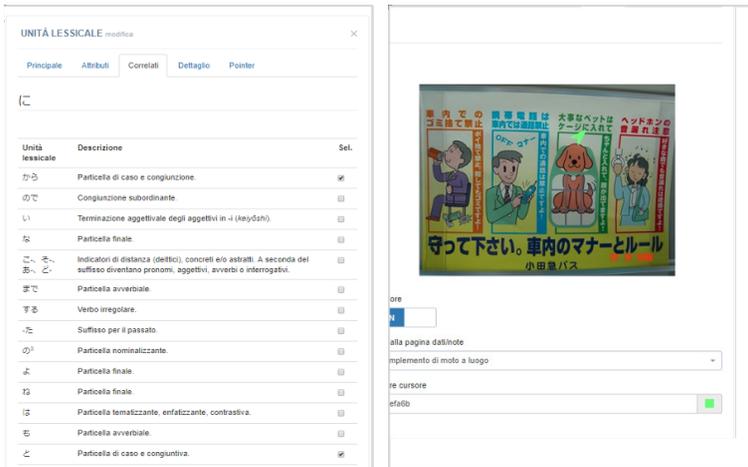
4 • Progettare un e-learning per il giapponese. Il case study *JaLea*

Figura 4.28 *JaLea*, area di backend. Dettaglio dell'interfaccia per la selezione degli elementi correlati e il posizionamento della freccia nell'immagine di descrizione

Le modifiche effettuate attraverso i tre tab ivi descritti appariranno automaticamente nella relativa pagina di frontend, come da schermata in figura 4.29. Rispettando il concetto basilare di manutenibilità, in *JaLea* non è più necessario inserire manualmente di volta in volta i dati relativi a immagini e video correlati, in quanto le miniature con il materiale multimediale vengono visualizzate automaticamente, in base al sistema di analisi che verifica la presenza dell'elemento grammaticale in esame in tutti i testi di video e immagini.

4 • Progettare un e-learning per il giapponese. Il case study *JaLea*

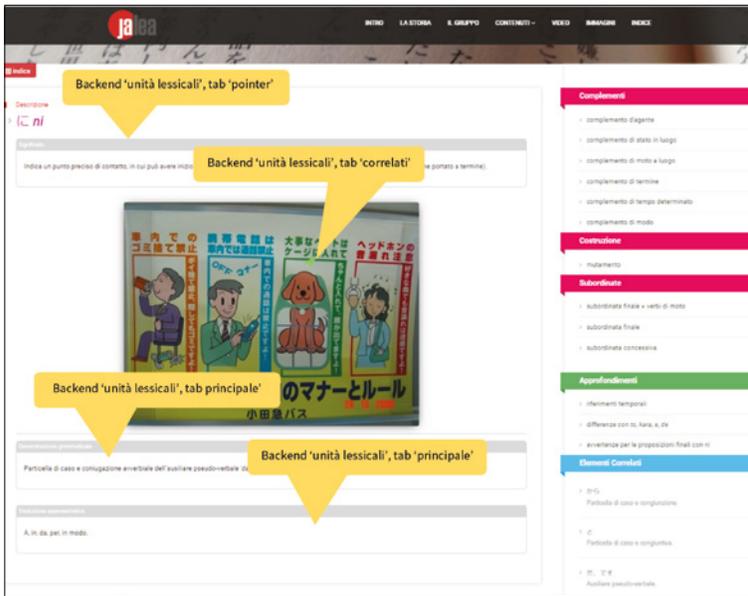


Figura 4.29 *JaLea*. Relazione tra l’inserimento delle informazioni nel backend e visualizzazione nell’area di frontend

Ulteriori dettagli del funzionamento del backend, soprattutto relativi alle modalità di creazione delle pagine di ‘Dettaglio’, e il loro collegamento alle pagine di ‘Descrizione’, verranno forniti nel capitolo 5, incentrato sull’analisi di *JaLea* attraverso un quadro di riferimento costituito dal framework di Garrett (2007) abbinato al modello di sviluppo ADDIE, in uso in ambito di instructional design.

4.6 *JaLea*: modalità di apprendimento attivo tramite esercizi

Una possibile evoluzione di *JaLea*, attualmente in fase sperimentale, riguarda la creazione di esercizi di *kanji* e grammatica. In base ai risultati di un'indagine effettuata nell'a.a. 2018-19, gli esercizi di *kanji* sono i più richiesti dai discenti di Ca' Foscari, soprattutto per rafforzare particolari competenze verificate anche in sede d'esame.

4.6.1 Motivazioni per la progettazione di un prototipo per l'apprendimento dei *kanji*

Come tutor incaricato del corso di lingua giapponese del primo anno del corso di Laurea magistrale in Lingue e civiltà dell'Asia e dell'Africa Mediterranea (LICAAM) nell'a.a. 2017-18, infatti, ho avuto la possibilità di analizzare i risultati degli esami scritti del 29 Agosto 2018, rilevando come il 48,5%³³ di coloro che avevano sostenuto l'esame aveva avuto numerose e molteplici difficoltà in una particolare prova, in cui avrebbero dovuto creare dei composti contestualizzati a partire da una lista di *kanji*.

In tabella 4.1, si riporta integralmente il test, che in base all'analisi quantitativa degli errori, è risultato particolarmente difficile per gli studenti.

Tabella 4.1 Test che richiede la selezione di *kanji* e la creazione di composti

<p>左の言葉と同じような意味を持つ三字の漢字熟語を()の中の漢字を使って作りなさい。 * lista di <i>kanji</i></p> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>識 値 非 然 健 自 能 常 性 価 可 康 観 不 的</p> </div> <p>例: 元気な様子: (健康的) な生活をすれば病気になるない。 1. 将来性がある様子: ヨーロッパの経済危機が回復する()はあまり期待できない。 2. 普通ではない変な様子: 友人に丁寧すぎる日本語を使うのも()に聞こえます。 3. 何かどのぐらい大切かという考え方やものの観方: あの人は()が全く違う。 4. 一般的な知識や判断力がないこと: 目上の方に対しあの言葉遣いは()だ。</p> <p>** Per ogni domanda la parte che precede i due punti [:] rappresenta la spiegazione, la parte che li segue è l'esercizio per cui si richiede di inserire nello spazio tra parentesi un composto di 3 <i>kanji</i> da selezionare dalla lista di cui sopra (*) *** Ogni <i>kanji</i> può essere selezionato una volta sola</p>

33 In particolare, le percentuali d'errore erano le seguenti: 1) Uno o più composti non presenti 30,1%; 2) Uno o più composti non corretti 18,4%.

Oltre all'analisi quantitativa, per conoscere opinioni e percezioni dei soggetti che non avevano superato l'esame, sono state proposte interviste qualitative non strutturate, svolte di persona e audio-registrate a cinque studenti che, oltre a frequentare gli incontri di tutorato, si sono anche resi disponibili a rispondere. L'intervista non strutturata ha permesso di evitare filtri affettivi e garantire un coinvolgimento reciproco tra intervistati e intervistatore ed ottenere quindi risposte il più possibile aderenti all'esperienza di studio vissuta.

Dalle interviste è risultato che la ragione per cui questo test aveva creato particolari problemi, era che l'unico modello disponibile per la preparazione nel libro di testo in uso, ovvero il *Bunka Chūkyū Nihongo* (BCN 2016, vol. 2), sarebbe stato in un formato differente da quello proposto nella prova d'esame. Quest'ultimo infatti richiedeva di selezionare da una lista di *kanji* un solo elemento, da aggiungere come prefisso o suffisso a composti di *kanji* già dati, e non di formare nuovi composti ex novo selezionando 3 *kanji* come richiedeva invece la prova di esame. Lo schema in tabella 4.2 riporta l'esercizio presente nel libro di testo (BCN 2016, 2: 213).

lista di <i>kanji</i>			
観 非 不 的			
自然 [] 親切 平等	常識 [] 公式 科学的	世界 印象 [] 本格	価値 人生 [] 宗教

Tabella 4.2 Estratto dell'esercizio di Pag.213 (BCN 2, 2016)

È stata proprio la mancanza della possibilità per gli studenti di fare esercizio su un modello simile a quello proposto nella prova di esame che mi ha suggerito di realizzare un sistema analogo all'interno di *JaLea*.

In fase di progettazione sono stati presi in considerazione i software *Moodle* (§ 2.1) e *Hot Potatoes*,³⁴ ma è stato rilevato che, sebbene sia possibile creare test a selezione multipla utilizzabili anche per fare esercizio di *kanji*, non si sarebbero potute creare regole che impedissero di utilizzare lo stesso carattere più di una volta, come avviene invece nel test di esame. Si è dunque deciso di realizzare un sistema ex-novo, con algoritmi ad hoc da inserire all'interno di *JaLea*.

Per quanto *JaLea* possa ritenersi un sistema derivato da *BunpoHyDict* infatti, bisogna sottolineare che è solo grazie alla attività di tu-

34 Applicazione Web che permette la creazione di esercizi di diverso tipo. Per dettagli si rimanda alla pagina del progetto: Half Baked Software, *HotPotatoes*: <https://hotpot.uvic.ca/>.

torato svolta in stretto contatto con gli studenti che si è potuto evolvere da sistema principalmente basato sulle competenze passive di lettura, ascolto e comprensione, a sistema e-learning in cui la parte attiva di ‘produzione’, seppur ancora limitata a esercitazioni, rende conto dei ripetuti feedback ricevuti durante molteplici presentazioni a convegni incentrate su *BunpoHyDict* e *JaLea*, dal 2008 ad oggi.

I cinque studenti che hanno accettato l’intervista, hanno poi collaborato attivamente anche alla creazione dei contenuti, attraverso la ricerca di materiale online, attività che avrebbe permesso loro anche di consolidare il proprio vocabolario di lingua giapponese.

Kolb (1984) e Krashen (1985), infatti, sottolineano come l’acquisizione sia fortemente legata alla partecipazione attiva del discente e alla consapevolezza di ciò che non si è ancora acquisito. Tale consapevolezza viene raggiunta tramite un’auto-analisi (Gardner, Miller, 1999) e lo studio delle proprie carenze, attività che si è quindi deciso di svolgere nel secondo periodo di tutorato, successivo agli esami della sessione estiva, nei mesi di settembre/novembre 2018.

Il materiale raccolto dagli studenti, elaborato e inserito in *Google Sheets*, è stato ovviamente sottoposto a correzioni. La selezione proposta dagli studenti infatti, riportava spesso frasi lunghe, eccessivamente complicate, o che suggerivano la soluzione in modo troppo esplicito.

La guida di questo periodo formativo ha quindi condotto ad avviare una nuova funzionalità di esercizi in *JaLea*, consentendo agli studenti collaboratori, così come ai futuri studenti utilizzatori, di rafforzare le proprie competenze relative ai *kanji*. Una seconda intervista, effettuata agli studenti coinvolti, questa volta dopo l’esame della sessione successiva ha evidenziato come questi avessero superato l’esame, e considerino tutt’oggi fondamentale l’attività svolta di ricerca dei contenuti da inserire negli esercizi dedicati alla memorizzazione di nuovi composti.

4.6.2 Realizzazione del prototipo

Considerando le logiche di sviluppo in un’ottica di instructional design, l’interfaccia per il prototipo del lato ‘Esercizi’ di *JaLea* è stata sviluppata valorizzando l’importanza del concetto di signifier descritto da Norman (2013).

Norman infatti utilizza il concetto di signifier (significante), come particolare segnalatore per colore, forma o proprietà della presenza dell’affordance (invito all’uso), o di una parte di questa (§ 3.1). Sono stati infatti creati differenti signifier grafici:

1. Piccoli riquadri in rilievo che richiamano la forma di un tasto da premere per selezionare i *kanji*.

2. Grandi riquadri colorati che evidenziano le definizioni a cui fare riferimento per dare la risposta esatta.



Figura 4.30 *JaLea*. Schermata iniziale della pagina ‘Esercizi’

3. Un’icona che, se cliccata propone, un riquadro informativo in sovrapposizione per segnalare la possibilità di ottenere informazioni sull’utilizzo del programma.

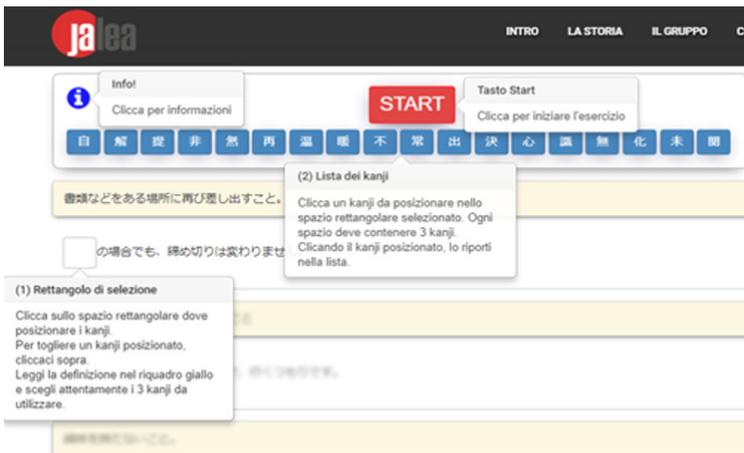


Figura 4.31 *JaLea*. Informazioni sull’utilizzo dell’esercizio

Seguendo un approccio di Interaction e instructional design orienta-

to all'attività cinestetica, piuttosto che fare apparire una pagina statica con istruzioni, si è preferito orientare immediatamente l'attenzione del discente sull'icona 'informazioni'. Cliccandola, si ottengono una serie di brevi informazioni sulle principali attività da svolgere ed è possibile svolgere prove di utilizzo fino a quando non si preme [Start]. Durante le prove di utilizzo però, tutti i testi, a parte quello della prima domanda, rimangono offuscati.

Questo tipo di presentazione 'semi-trasparente', si basa su una metodologia di organizzazione della presentazione delle informazioni a scopo didattico, relativamente recente rispetto alle prime applicazioni delle 9 regole di instructional design definite da Gagnè in *The Conditions of Learning* (1965) trasposte in ambiente all'e-learning. Infatti, mentre negli anni Settanta si offrivano semplici schermate introduttive ricche di testo, nel nostro caso di studio si obbliga lo studente a interagire immediatamente con l'interfaccia, e a ricavarne informazioni immediate.³⁵

Premendo il tasto [START] il sistema rimescola le domande in modo casuale, le visualizza in sequenza, e avvia il conto alla rovescia. Si è infatti cercato di limitare il carico cognitivo dello studente, e di ottimizzare i tempi di utilizzo dell'interfaccia attraverso due strategie: una visiva e una operativa.

La strategia visiva si basa sulla continua visualizzazione dell'area contenente il cronometro e la lista dei *kanji* da selezionare, in qualsiasi momento, anche in caso di scorrimento della pagina, così come avviene nelle più recenti versioni di *Moodle*.

La strategia operativa si basa più semplicemente sulla implementazione di un sistema che consente di selezionare prima la sezione/casella su cui operare, e successivamente il *kanji* da inserire attraverso un singolo click, senza utilizzare 'trascinamento e rilascio' (drag & drop).

Ogni volta che lo studente clicca un *kanji* della lista, questo scompare e viene posizionato nella casella rettangolare preselezionata. Cliccando nuovamente sul *kanji* posizionato, invece, questo viene riportato indietro nella lista.

Attraverso la strategia visiva, quindi, si permette allo studente di avere continuamente sotto controllo il tempo a disposizione e i *kanji* da selezionare, aiutando la concentrazione; attraverso la strategia operativa, inoltre, si risparmia tempo rispetto al movimento drag & drop, e si risolvono le anomalie relative ai dispositivi mobili, dove il trascinamento degli elementi attraverso le dita non è sempre preciso.

35 Si veda a questo proposito i concetti di *Hands-on* e *Advice* di Tom Kuhlmann (2014).



Figura 4.32 *JaLea*. Istantanea di un test iniziato. Cronometro al posto del tasto ‘start’, barra costante dei *kanji* selezionabili, riquadro giallo con quelli selezionati, due sezioni diverse per risultato corretto o errato

Per verificare se questo tipo di funzionalità fossero apprezzate dagli studenti che utilizzano *JaLea*, ho proposto loro il questionario analizzato nel capitolo 6.

Le attività di studio dello stato dell’arte e conseguente ricerca e sviluppo riportate fin qui, hanno condotto alla creazione di un quadro teorico di riferimento, descritto nel capitolo 5 e successivamente applicato ad alcune funzionalità di *JaLea*, per evidenziare quali strategie sono state effettivamente utilizzate per creare l’esperienza utente’ vissuta dal discente in rapporto all’artefatto digitale.

