Corso di Tecnologie Assistive per la Didattica A.A. 2017

LABORATORIO

Progettazione/Sviluppo di Web APP accessibili e usabili a supporto della disabilità

Ing. Caterina Senette
Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche

COSA SONO LE WEB APP

ESEMPI DI WEB APP GENERICHE

EVERNOTE

Esplora Web Clipper Scarica l'app

Accedi Registrati

Prova Evernote Business

Ricorda tutto

Organizza il tuo lavoro e riordina la tua vita. Raccogli tutto ciò che è importante per te in un sol posto e trovalo quando ti serve, velocemente.

Crea il tuo account gratuito

Lavora in modo più intelligente

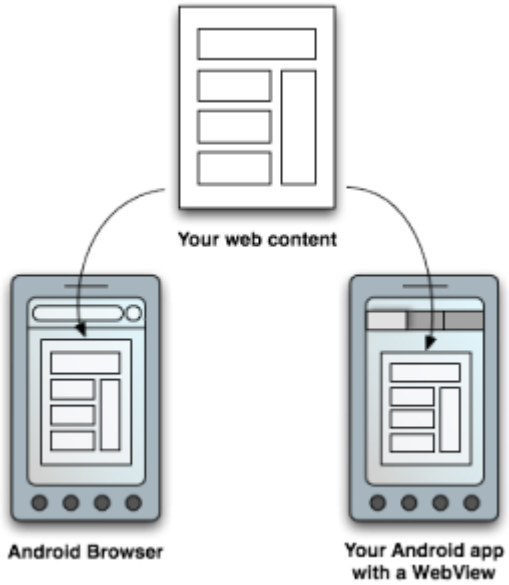
Migliora le tue note con link, liste di controllo, tabelle, allegati e registrazioni audio. Sono ricercabili anche le note scritte a mano.

Crea il tuo account gratuito

APP NATIVE

- Il lavoro è eseguito dal device
- Possono essere più responsive
- Hanno più performance in termini di interattività
- Non dipendono dalla rete Internet

- Devono essere sviluppate specificamente per la piattaforma (Android, iOS, Windows) di destinazione.
- Vanno scaricate e installate tramite lo store specifico del sistema operativo
- Ogni aggiornamento richiede un nuovo download

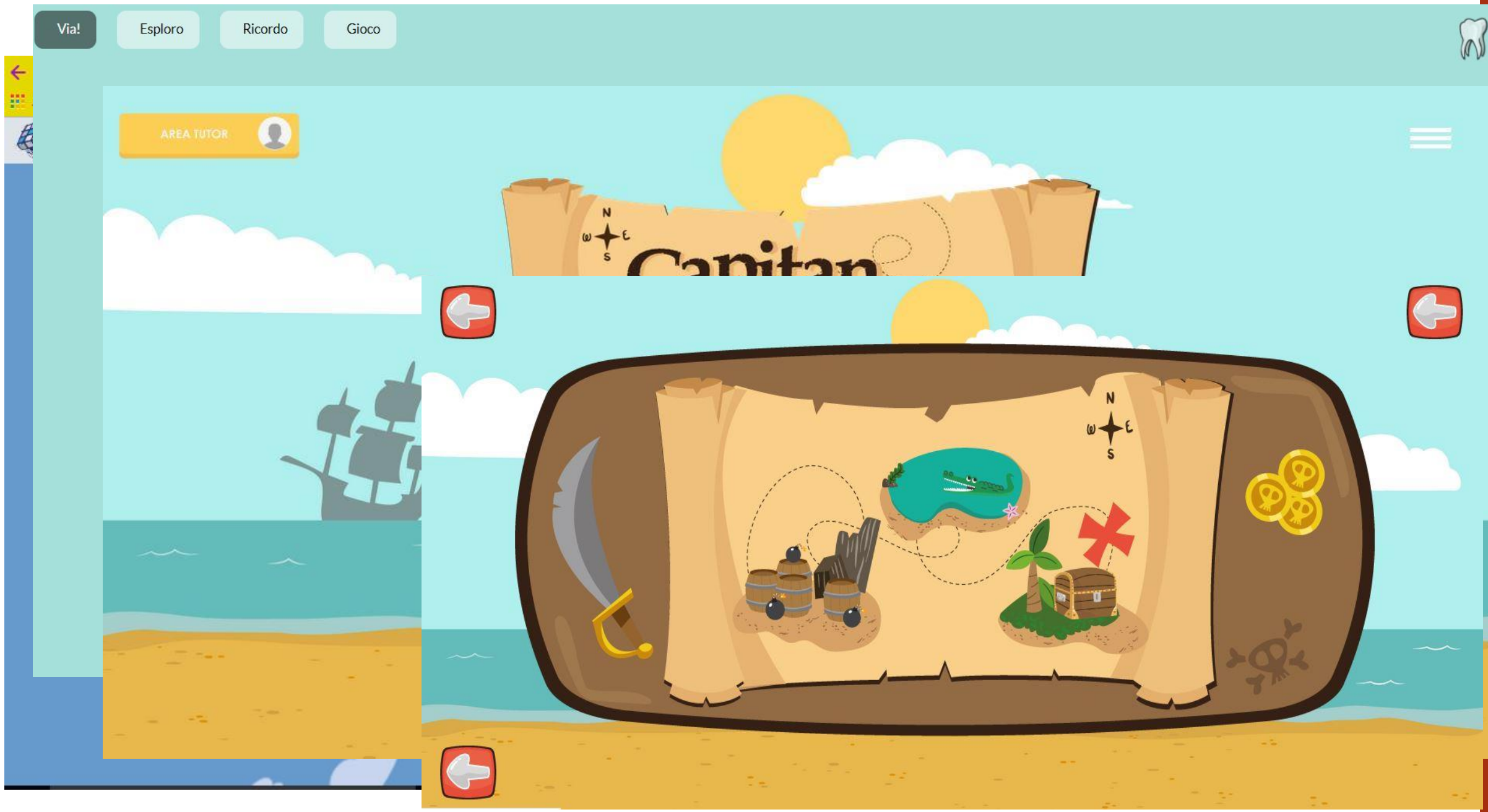


WEB APP

- Il grosso del lavoro è svolto dal server
- Sono accessibili tramite il browser del proprio dispositivo, non necessitano di essere scaricate e installate
- Possono essere scritte in qualsiasi linguaggio piaccia allo sviluppatore, purché il prodotto finale consegnato al browser del client sia **HTML5+CSS3+Javascript**

- Non c'è necessità di installazione e aggiornamento perché la pagina cui l'utente accede è sempre l'ultima versione disponibile.
- Lo sviluppatore dovrà fare attenzione alle **differenze di comportamento fra i diversi browser**
- **Non si ha accesso completo all'hardware** (in alcuni casi questo potrebbe essere un punto a favore…) e pertanto alcune funzionalità potrebbero essere precluse.
- Potrebbero essere **meno performanti e responsive**

ESEMPI WEB APP SPECIFICHE PER LA DIDATTICA ASSISTIVA


Via!
Esploro
Ricordo
Gioco
AREA TUTOR
N
W
E
S
Capitan

INTRODUZIONE TEORICO-PRATICA ALLA REALIZZAZIONE DI WEB-APP ACCESSIBILI/USABILI

SYLLABUS

- Preparazione ambiente di sviluppo
- Progettare la struttura dell'applicazione
- Struttura per la raccolta dati

SYLLABUS

Preparazione ambiente di sviluppo

Cosa serve per lo sviluppo di una WebAPP

LATO SERVER

Macchine - LINUX
PHP engine
Database Engine → MySQL
Server Web → Apache..

LATO CLIENT

HTML + HTML 5
JQUERY E JSON
AJAX
RAPHAEL
…

Chi progetta applicazioni Web si serve di **almeno 2 ambienti server** e di quanti più possibile **ambienti Client** per la verifica Cross-Browser

- Server locale per l'implementazione e il test:
  Xampp
  EasyPhp ecc..
  dipende anche dal S.O. della propria macchina locale..

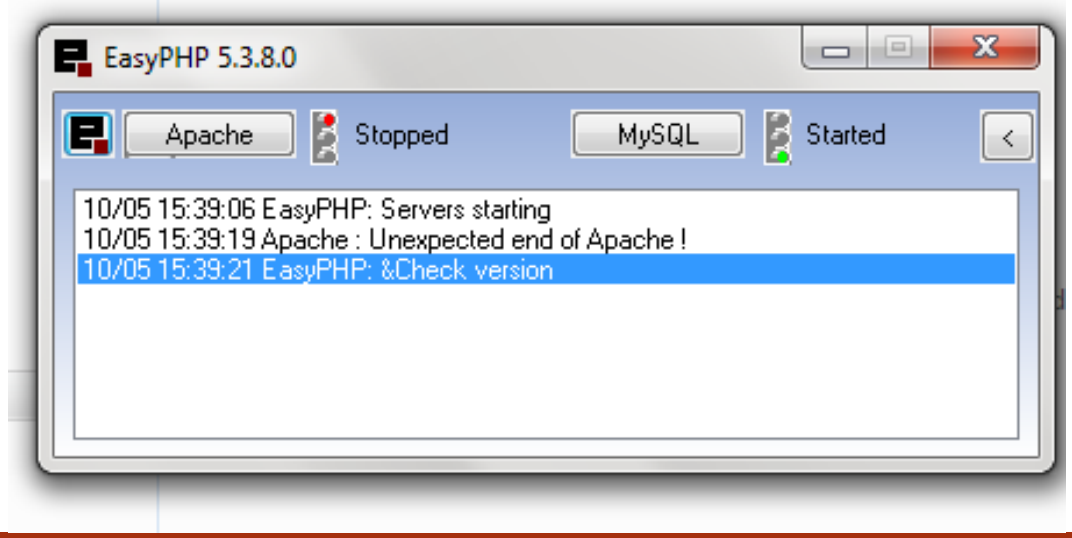
  Sono pacchetti SW di tipo WAMP comprendenti: PHP 5, un web Server: Apache 2, un SQL Server: MySQL 5

- Server remoto per il "deployement reale" dell'applicazione a indirizzo IP pubblico

SERVER LOCALE

- Server locali: hanno un'interfaccia d'amministrazione che permette di gestire gli utenti, l'avvio e lo spegnimento dei server.

- Il server Apache crea automaticamente di default un dominio virtuale (in locale) all'indirizzo di localhost (http://127.0.0.1).



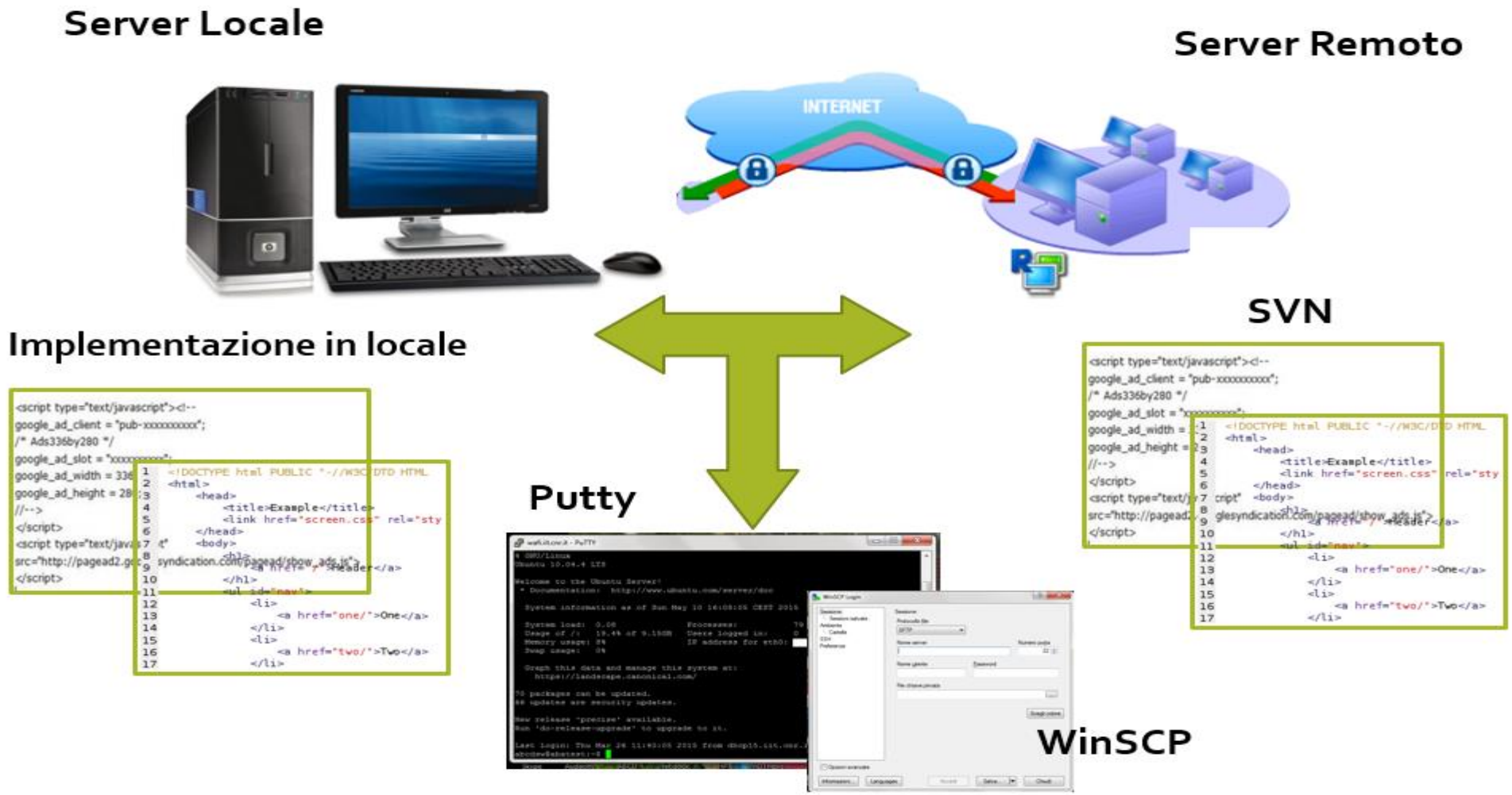
Client == Browser

Il punto nodale in un'applicazione Web è il **Browser** dove l'applicazione viene fruita:

- I diversi Browser supportano diversamente le varie tecnologie Internet
- Il mondo delle piattaforme **MOBILI** pone sfide sempre maggiori

L'ideale sarebbe una programmazione **cross-browser** ma non sempre questo è possibile a costo zero …

COMUNICAZIONE CLIENT-SERVER



SYLLABUS

Progettare la struttura dell’applicazione

Per disegnare la STRUTTURA di una qualunque APPLICAZIONE WEB (e non) occorre:

- CONOSCERE IL DOMINIO
- CONOSCERE GLI USE- CASES
- CONOSCERE IL TARGET UTENTE (i target se ce ne sono più di uno) e le condizioni di UTILIZZO nell'ambito del dominio

Conoscere il DOMINIO serve a:

- Capire lo SCOPO dell'applicazione (ludico/supporto all'apprendimento ecc.)
- Capire la frequenza di accesso, la presenza o meno di accessi concorrenti, tempo di risposta accettabile
- Il tipo o le tipologie di dispositivi HW che l'utente userà per accedere
- Capire come impostare la STRUTTURA DATI (dati di funzionamento e dati eventuali da collezionare)
- Capire il CONTESTO ambientale (fisico e psicologico) in cui l'utente potrebbe fare uso dell'applicazione

Conoscere gli Use-Cases serve a:

- Capire QUALI tipologie di utenti useranno l'applicazione (per prevedere diverse risposte in base al profilo)

- Capire QUANTI utenti la useranno per stimare la frequenza di accesso, la presenza o meno di accessi concorrenti, tempo di risposta accettabile

- Disegnare opportunamente il flusso di navigazione in modo che l'utente raggiunga sempre il suo scopo, ad ogni interazione, con il COSTO minore

- Capire come impostare la STRUTTURA DATI

Conoscere il TARGET UTENTE significa predisporsi a:

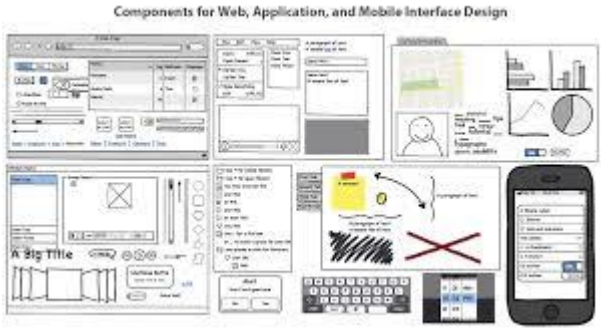
Disegnare/Progettare **CON l'utente** la quale cosa, soprattutto nei casi di Utenti Speciali, permette di:

- **Facilitare l'interazione** a livello di dispositivi di input/output
- **Semplificare la navigazione**
- **Diversificare gli output** dell'applicazione in modo accessibile ai diversi tipi di disabilità
- Rendere **sicura** la navigazione
- **Motivare l'utente** a non abbandonare l'uso dell'applicazione

Strumenti utili alla progettazione

- Il disegno **PARTECIPATIVO**

- **WIREFRAMES o MOCKUPS**



- I **Test di Usabilità** da fare **durante** la progettazione/implementazione

Esempio 1: MACRO struttura di **CLG**

Cognitive Learning Games

TUTOR

- Registrazione
- Inserisce/gestisce i suoi studenti

Gestisce RISORSE

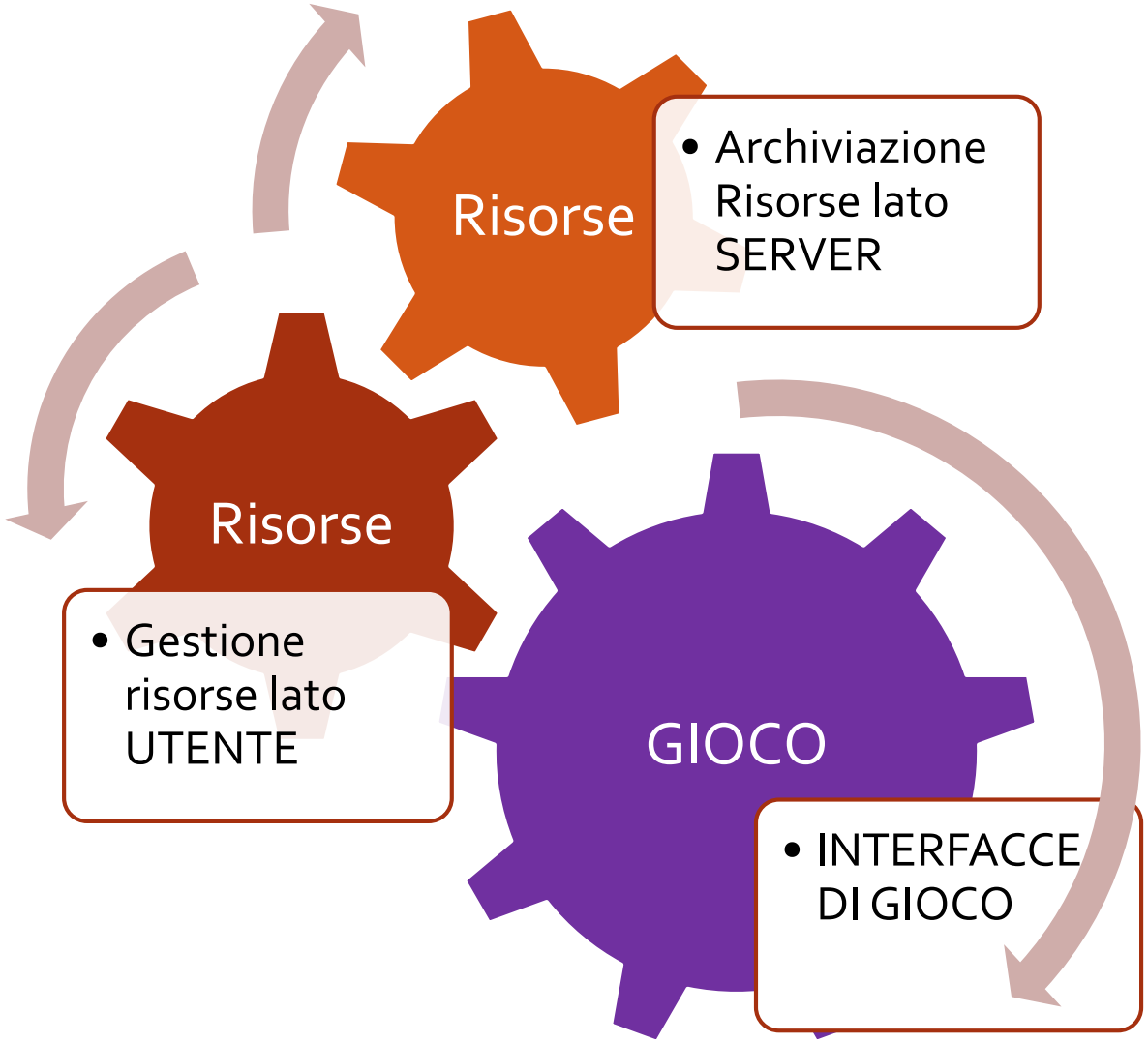
- Crea nuovi giochi
- Inserisce nuove Immagini/Video
- Li associa ad ogni studente

Learning Analytics

- Verifica i progressi nel tempo

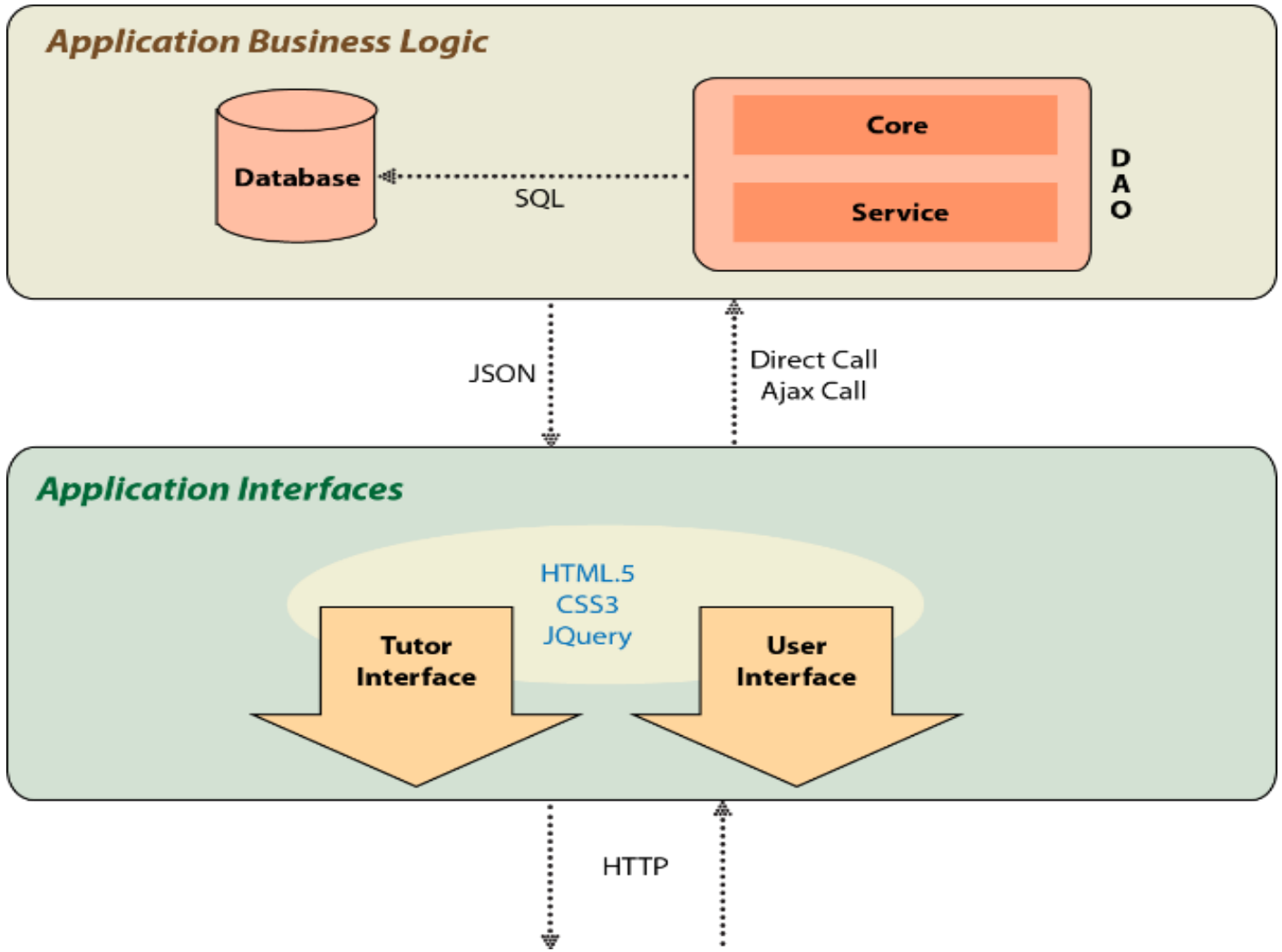
Cognitive Learning Games

Esempio 1: struttura di **CLG**



Cognitive Learning Games

Esempio 1: struttura LOGICA di **CLG**



Esempio 1: MACRO struttura di **ABCD SW**

TUTOR

Registrazione	Inserisce/gestisce i suoi studenti	Definisce il PROFILO dello studente

Gestisce RISORSE

Inserisce nuove Risorse Immagini	Le associa ad ogni studente

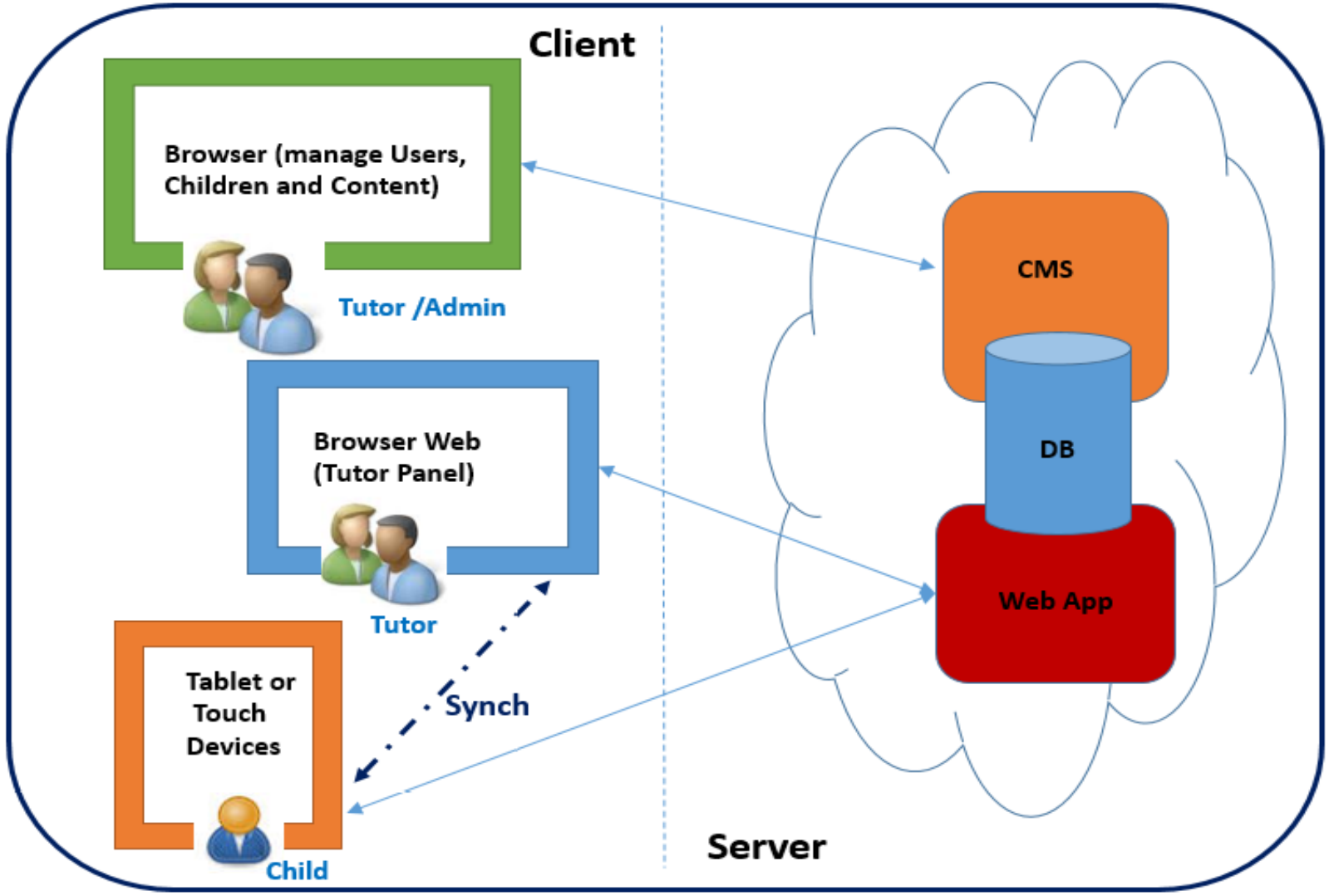
CONTROLLA IL FLUSSO DI GIOCO IN TEMPO REALE

Imposta il PROGRAMMA/LIVELLO	Inserisce la valutazione

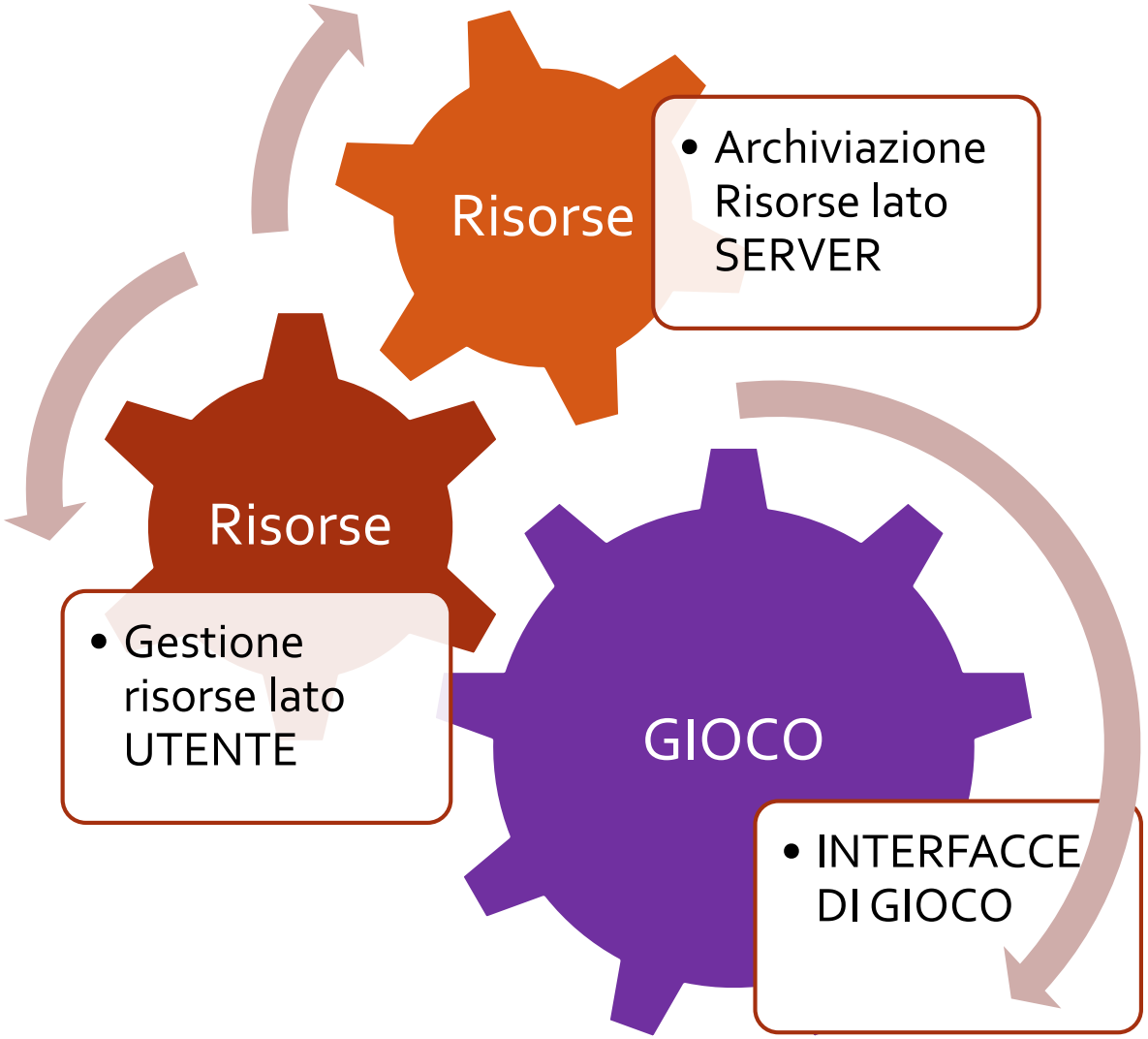
Learning Analytics

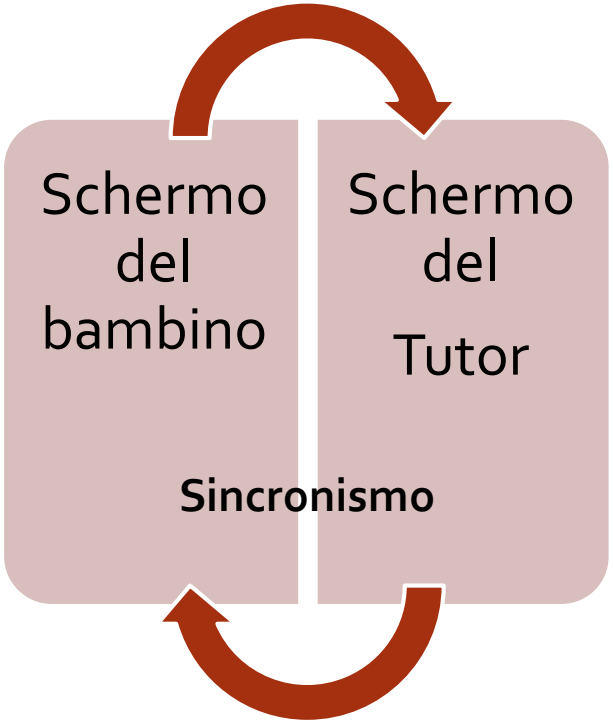
Verifica i progressi nel tempo

Esempio 2: struttura di **ABCD SW**



Esempio 2: struttura di **ABCD SW**





Rispetto a una classica Applicazione WEB nel nostro caso la progettazione deve prevedere 3 funzionalità particolari

1. **Interazione con le risorse** tramite GIOCHI (CLG)
2. **Controllo del flusso di gioco** da parte del Tutor: sincronizzazione (caso di ABCD)
3. **Statistiche di gioco**, in tempo reale e a posteriori tramite storico (ABCD/CLG)

SYLLABUS

Struttura per la raccolta dati

Progettare la BASE DATI (MySQL)

Nella progettazione di un database è fondamentale tenere presente alcune caratteristiche basilari:

1. Il Database va progettato PRIMA dell'applicazione e NON il contrario
2. Vanno definiti gli SCOPI dell'applicazione (esigenze operative/esigenze decisionali): il tipo di dati da collezionare e l'interazione con essi (archiviazione, continua interrogazione/ elaborazione ad alto livello ecc.)

Fatto questo si procede a costruire un modello della realtà di studio individuando:

- Il numero delle tabelle che verranno coinvolte dalle interrogazioni lanciate tramite l' applicazione
- Il numero e il tipo di campi da creare all'interno delle tabelle
- Le relazioni interne ed esterne tra i campi

Progettare la BASE DATI(MySQL)

Modello Entità/Relazione (E/R)

1° fase

- Analisi delle caratteristiche dei dati;
- Analisi delle operazioni da effettuare sui dati;
- Analisi degli eventi che possono influenzare i dati;
- Analisi dei vincoli d'integrità, cioè delle proprietà relative a informazioni ed eventi.

Progettare la BASE DATI(MySQL)

Modello Entità/Relazione (E/R)

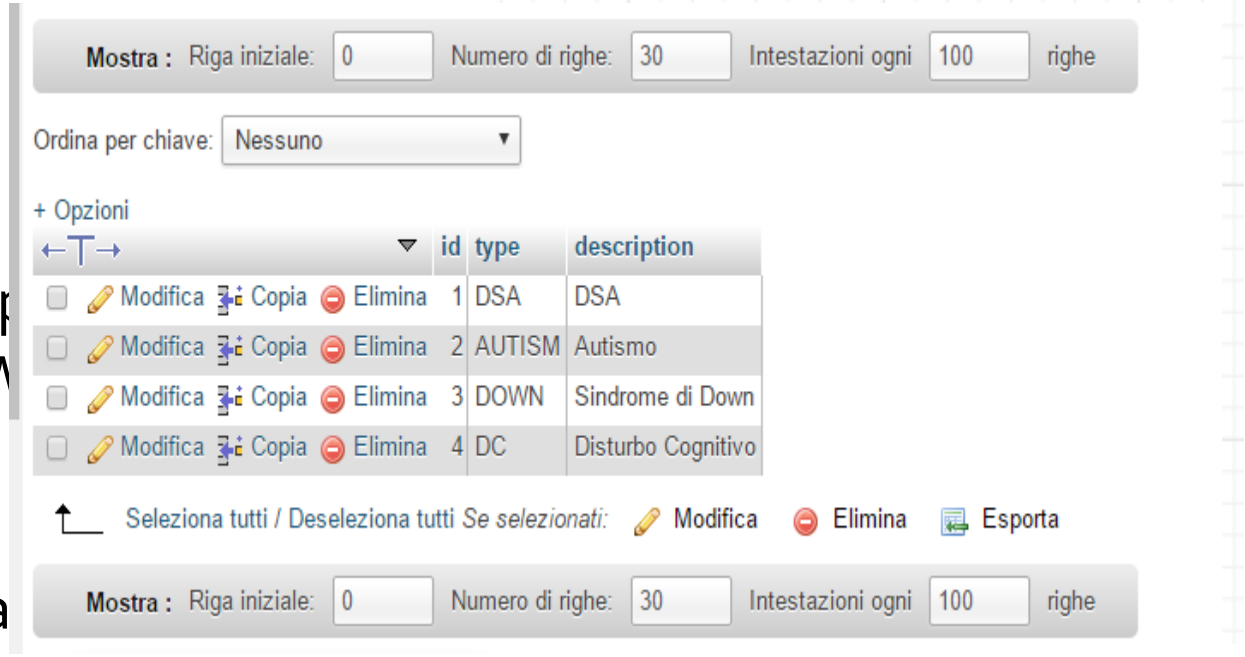
2° fase

- Definire **quali tabelle** dovranno essere create
- Quali **relazioni** dovranno sussistere **tra** di esse
- A quali **vincoli** saranno sottoposte le relazioni

Progettare la BASE DATI(MySQL)

Cognitive Learning Games

Quali sono i metodi per **costruire** le tabelle e per **popolarle**?
Pre-requisito: avere già chiara la struttura delle **entità** e **le relazioni**



COSTRUIRE

- Usando script MySQL
- A mano, usando per esempio php
- Usando altri Tool come MySQL W

POPOLARE

- A mano solo per le tabelle che va volta per tutte (esempio «diagnosis»)
- Da codice per tutte le altre

Nella pratica…

Per piattaforma WINDOWS

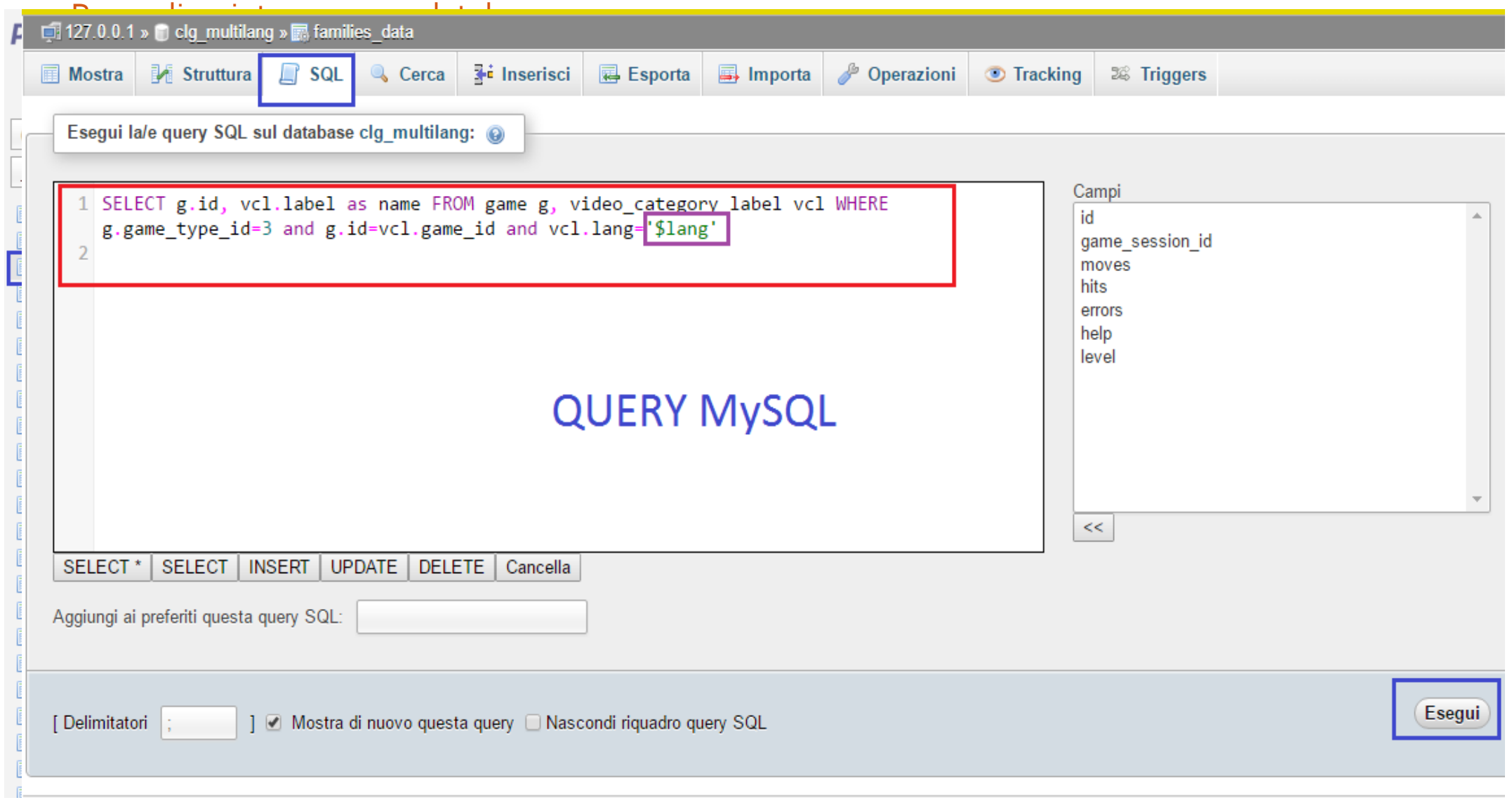
- ❑ Scaricare EASYPHP 14.1 da http://www.easyphp.org/easyphp-webserver.php
- ❑ Scaricare ECLIPSE IDE per sviluppo su piattaforma PHP da https://eclipse.org/downloads/eclipse-packages/
- ❑ Scaricare WINSCP
- ❑ Scaricare PUTTY
- ❑ Scaricare TORTOISE SVN

Per piattaforma MAC OS X

- Scaricare MAP da https://www.mamp.info/en/
- Scaricare ECLIPSE IDE per sviluppo su piattaforma PHP da https://eclipse.org/downloads/eclipse-packages/
- Scaricare FileZilla da https://filezilla-project.org/
- Usare Terminal per la connessione *ssh*
- Alternative a TORTOISE SVN http://formac.informer.com/tortoisesvn

:: *Progettare la STRUTTURA DATI*

- Impariamo a usare phpMyadmin



Capitolo 2

Introduzione teorico-pratica all'uso delle principali TECNOLOGIE WEB (librerie) operanti nelle Web-APPs

SYLLABUS

- Principali tecnologie diffuse nelle Web App
  - AJAX/JSON
  - JQUERY
  - HTML5
  - HIGHCHART
- Strategie di sviluppo

Regole di buona programmazione

1. Esistono N modi per implementare una stessa funzionalità io vi mostro 1 degli N modi

2. Imparare a distinguere la struttura della pagina HTML dallo specifico layout che va reso usando sempre i CSS.

3. Separare struttura e layout dalle funzionalità che esprime la pagina e che preferibilmente risiedono in un file di codice a parte solitamente JAVASCRIPT (.js)

4. Capire come rispondere agli EVENTI della pagina: caricamento pagina/click del bottone ecc.

Regole di buona programmazione

5. Le nostre pagine sono sempre pagine .PHP o .JS. Imparare a distinguere le due tipologie e a mettere in comunicazione i 2 linguaggi

6. Assegnare ai file nomi coerenti con le pagine che rappresentano usando preferibilmente la notazione CAMEL

6. Separare sempre la gestione del DATABASE di riferimento per una data applicazione scrivendo tutto il codice necessario per la connessione (apertura/chiusura) chiamate(Queries) in scrittura e in lettura ecc. in file a parte

Richiami TEORICI

1. JQUERY

Un framework sviluppato con il preciso intento di rendere il codice più sintetico e di **limitare al minimo l'estensione degli oggetti globali** per ottenere la massima compatibilità con altre librerie.
È dunque una libreria per la manipolazione

- **Degli stili CSS e degli elementi HTML**
- **Degli effetti grafici**
- **Dei metodi per chiamate AJAX cross-browser**

Il tutto senza toccare nessuno degli oggetti nativi JavaScript. Tutto ruota attorno all'oggetto/funzione $

```
$("#mioBlocco"); //Un oggetto jQuery
jQuery("#mioBlocco") //Lo stesso oggetto richiamato in modo diverso
```

2. AJAX

- Permette lo sviluppo di APP Web basate su uno scambio di dati in background fra web browser e server, che consente **l'aggiornamento dinamico di una pagina web** senza esplicito ricaricamento da parte dell'utente.
- **AJAX è asincrono** nel senso che i dati extra sono richiesti al server e caricati in background senza interferire con il comportamento della pagina esistente.
- Normalmente **le funzioni richiamate sono scritte con il linguaggio JavaScript**. Tuttavia, e a dispetto del nome, l'uso di JavaScript e di XML non è obbligatorio, come non è detto che le richieste di caricamento debbano essere necessariamente asincrone.

Richiami TEORICI

3. JSON

In informatica, nell'ambito della programmazione web, JSON, acronimo di JavaScript Object Notation, è un formato adatto all'interscambio di dati fra applicazioni client-server.
È basato sul linguaggio JavaScript Standard ECMA-262 3ª edizione dicembre 1999, ma ne è indipendente. Viene usato in AJAX come alternativa a XML/XSLT.
http://www.json.org/json-it.html

4. HTML5

- Le novità introdotte dall'HTML5 rispetto all'HTML4 sono finalizzate soprattutto a migliorare il **disaccoppiamento fra struttura**, **definita dal markup, caratteristiche di resa (tipo di carattere, colori, eccetera), definite dalle direttive di stile, e contenuti di una pagina web, definiti dal testo vero e proprio.**
- Inoltre l'HTML5 prevede il supporto per la memorizzazione locale di grandi quantità di dati scaricati dal web browser, per consentire l'utilizzo di applicazioni basate su web (come per esempio le caselle di posta di Google o altri servizi analoghi) anche in assenza di collegamento a Internet.

Richiami TEORICI

3. HIGHCHART

- Libreria per la costruzione di grafici interattivi su applicazioni Web interamente basata su JAVASCRIPT e sulle tecnologie Web native, non richiede plug-in.

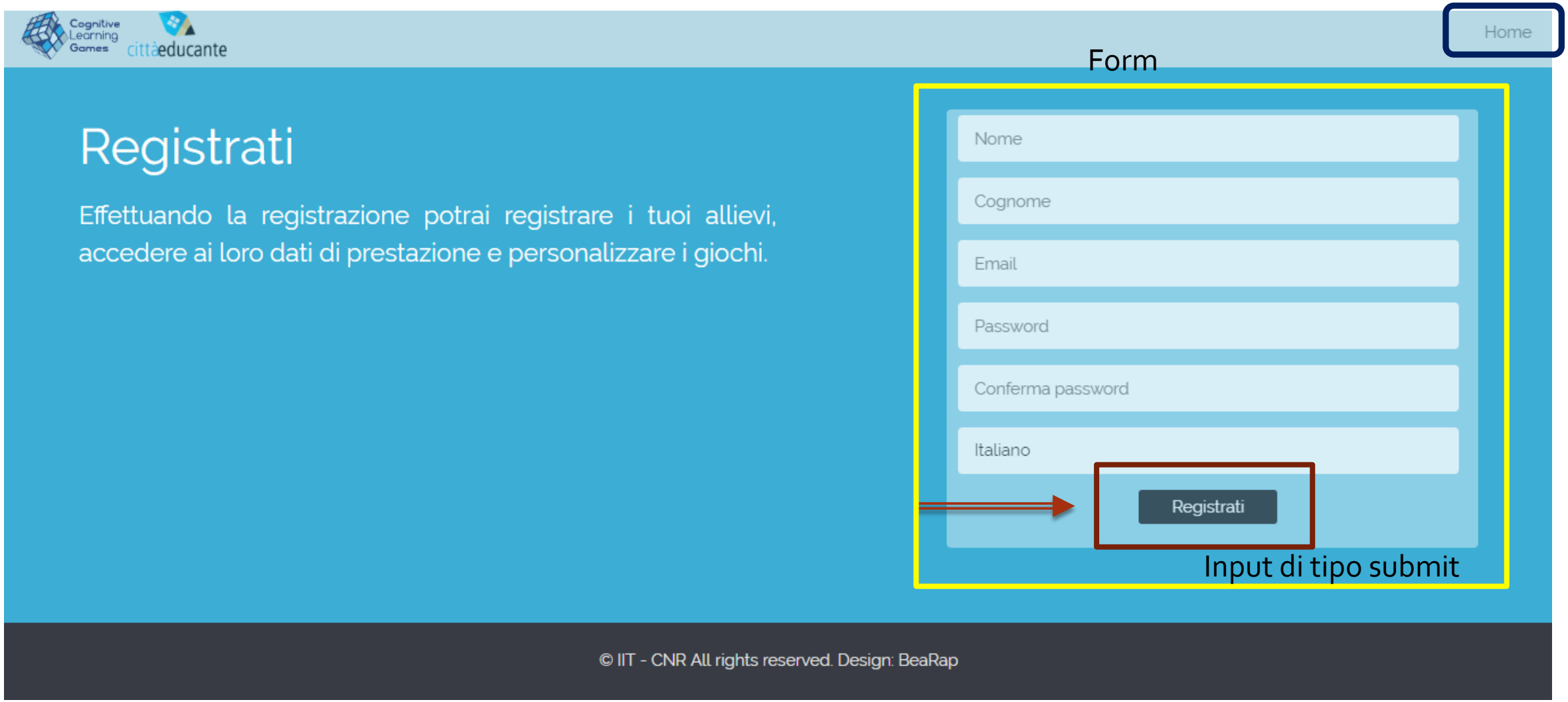
- Funziona in tutti i moderni browser mobili e desktop, tra cui iPhone/iPad e Internet Explorer a partire dalla versione 6. Su iOS e Android il supporto multitouch fornisce un'esperienza utente senza soluzione di continuità.

- Il rendering grafico finale dipende dal browser, quelli standard utilizzano SVG, Internet Explorer usa VML.

Funzionalità tipiche

1) REGISTRAZIONE E GESTIONE UTENTI

Origine pagina



1.GARANTIRE IL MULTILINGUISMO CATTURANDO L'INFO SULLA LINGUA DELL'UTENTE DALLA PAGINA HOME DI PARTENZA

```
$lang = "it";
if(isset($_REQUEST['lang']))
$lang = $_REQUEST['lang'];
```

2.SPECIFICARE IL COMPORTAMENTO DELLA PAGINA AL SUO CARICAMENTO

```
$(document).ready(function() {
.....
.....
initRegistrationPage();
....
});
```

3.DEFINIRE E GESTIRE IL COMPORTAMENTO DELLA FORM CHE CATTURA I DATI DI INPUT DELL'UTENTE

Nel nostro caso il comportamento della form è legato alla definizione del SUBMIT e al controllo da associare ai campi della form secondo regole precedentemente decise ex: ***testEmail()***

4.GESTIRE LA SOTTOMISSIONE DEI DATI INSERITI DALL'UTENTE CHE PREME IL TASTO DI SUBMIT

```
<input id="submit-button" value="<?php write_s("reg-button-registra", $lang)?>" class="button special" type="button" onclick="registerUser()">
```

Andiamo a vedere la funzione function registerUser() direttamente sul file csg.js

Elementi della funzione:

- **Imposta la variabile targetUrl per la chiamata AJAX:**
  - **var targetUrl = "dao/service.php?action=registerUser";**
- **Definisce le variabili che catturano i dati della FORM**
- **Controlla la presenza di campi nulli, se sono nulli restituisce un ALERT**
- **Controlla la coerenza dell'indirizzo e-mail (deve essere unico per ogni utente)**
- **Ad ogni passo arricchisce la variabile targetUrl con i nuovi dati**
- **Imposta la chiamata AJAX**

```
$.ajax({
    url : targetUrl,
    type : "POST",
    async : true,
    dataType : "json",
    contentType : "application/json; charset=utf-8",
    success : function(data, stato) {
        if (typeof data.error != 'undefined') {
            if (data.error.indexOf("already registered") > 0) {
                message = jStrings[jsLang]['reg-already-reg-1']
                        + nickname
                        + jStrings[jsLang]['reg-already-reg-2'];
                ....
            }
        } else {
            message = jStrings[jsLang]['reg-registered-1']
                    + nickname
                    + jStrings[jsLang]['reg-registered-2']
                    + nickname
                    + jStrings[jsLang]['reg-registered-3'];
            $("#name").css("color", color);
            ...
            $("#go-tutor").show();
        }
    },
    error : function(richiesta, stato, errori) {
        alert("Unexpected Error");
        console.log("Unexpected Error [function: registerUser()]");
    }
});
```

impostazioni

successo

errore

Il lavoro finale spetta alla QUERY che inserisce finalmente i dati nel DATABASE
e che viene chiamata da `$dao->addUserData($user);`

```
public function addUserData($user) {
    $tutorId = "NULL";
    if ($user['tutor'] != FALSE) {
    $tutorId = $user['tutor'];
    }
    $query = "insert into users (user_role_id,
    nickname, password, tutor_id) values (";
    $query .= "(select id from user_role where
    role= '" . $user['role'] . "'), ";
    $query .= "'" . $user['name'] . "', ";
    $pass = ($user['pass'] != FALSE) ? "MD5('" .
    $user['pass'] . "')" : "NULL";
    $query .= "$pass, ";
    $query .= "$tutorId)";
    return $this->doQuery($query, "createUser");
}
```

MULTILINGUISMO

Sia CLG che ABCD SW sono software multilingua. Il loro multilinguismo è definito da codice e necessita di:

- Traduzione in una data lingua di tutte le stringhe di testo presenti nelle interfacce del SW
- File audio registrati nelle varie lingue (per gli esercizi che prevedono PROMPT vocale)

Per rendere tutto molto semplificato esistono **dei vettori definiti da codice**, del tipo **chiave/valore** che contengono tutte le stringhe in una certa lingua, 2 vettori per ogni lingua

- **Uno per le stringhe di testo richiamate con JAVASCRIPT**
- **Uno per le stringhe di testo richiamate da PHP**

Una funzione PHP viene lanciata all'accesso dell'utente al SW. Una volta letto (da QUERY MySQL) il dato linguistico associato all'utente in fase di registrazione questo dato è passato alla funzione che rende disponibili tutti i campi di quel vettore che da quel momento sono richiamati ogni qual volta l'interfaccia preveda un'etichetta testuale

Andiamo a vedere il file **strings.php** nella cartella dell'applicazione. Come dice la stessa intestazione:

```
* $_STRINGS: contiene le stringhe gestite dal PHP
* $_STRINGS_JS: contiene le stringhe gestite da JAVASCRIPT
```

```
$_STRINGS = array (
    "it" => array (
    "home-play" => "Gioca",
    "common-close" =>
    "Chiudi",
    …..
    ),
    "en" => array (
    "home-play" => "Play",
    "common-close" => "Close",
…
));
```

```
$_STRINGS_JS = array (
    "it" => array (
    "common-level" => "Livello",
    "login-err-msg-1" => "Il campo \"Nome\" deve contenere un nome
    valido.",…..
    ),
    "en" => array (
    "home-play" => "Play",
    "common-close" => "Close",
…
));
```

Utilizzare la funzione *write_s()* per inserire le stringhe da PHP

Utilizzare la funzione *toJavascript()* per creare l'array 'jStrings' di stringhe JAVASCRIPT all'interno della pagina

```
function write_s($label, $lang = NULL) {
    global $_STRINGS;

    if (isset($_SESSION['lang'])) {
    echo
    $_STRINGS[$_SESSION['lang']][$label];
    }
    else if ($lang != NULL) {
    echo $_STRINGS[$lang][$label];
    }
    else {
    echo $_STRINGS['it'][$label];
}
}
```

```
function toJavascript() {
    global $_STRINGS_JS;

    echo "var jStrings = {\n";
    foreach ($_STRINGS_JS as $country => $strings) {
    echo " '" . $country . "' : {\n";
    foreach ($strings as $label => $text) {
    echo "  '" . $label . "' : " . '"' . $text . '",\n';
    }
    echo "  'dummy-dummy' : 'dummy-dummy'\n";
    echo " },\n";
    }
    echo " 'DUMMY' : {}\n";
    echo "};\n"; }
```

Utilizzare la funzione *write_s()* per inserire le stringhe da PHP

Utilizzare la funzione *toJavascript()* per creare l'array 'jStrings' di stringhe JAVASCRIPT all'interno della pagina

ESEMPI

```
<select id="combo-groupby"  class="filter">
    <option id="group3" value="group3"><?php
write_s("chart-resolution-3")?></option>
    <option id="group1" value="group1"><?php
write_s("chart-resolution-1")?></option>
        <option id="group2"
value="group2"><?php write_s("chart-resolution-
2")?></option>

</select>
```

Nel file php che mi definisce l'html dell'interfaccia faccio agire la funzione **toJavascript ()** che è una funzione php (quindi la esegue il server)

La funzione scorre tutti gli elementi della stringa **$_STRING_JS** e li carica in un array **jString** all'interno della pagina in cui è chiamata la funzione

La singola stringa di testo che JS deve usare è richiamata semplicemente da

jStrings[jsLang]['chiave_X']

in base al valore di jsLang

STRATEGIE DI SVILUPPO

Strategie di SVILUPPO

STRUMENTI DI SVILUPPO DEL BROWSER

- Velocizzano lo sviluppo, il test e il debug delle pagine web
- Sono disponibili nei browser più diffusi anche se con funzionalità diverse.
- Si accede tramite
  - F12: Internet Explorer, Googe Chrome, Mozilla Firefox
  - CTRL + MAIUSC + I: Opera, Google Chrome
  - CTRL + ALT + I: Safari
  - Tasto destro – ispeziona elemento

Strategie di SVILUPPO

STRUMENTI DI SVILUPPO DEL BROWSER

1. Andiamo nella pagina dell'applicazione
   wafi.iit.cnr.it/stella2/clg/index.php

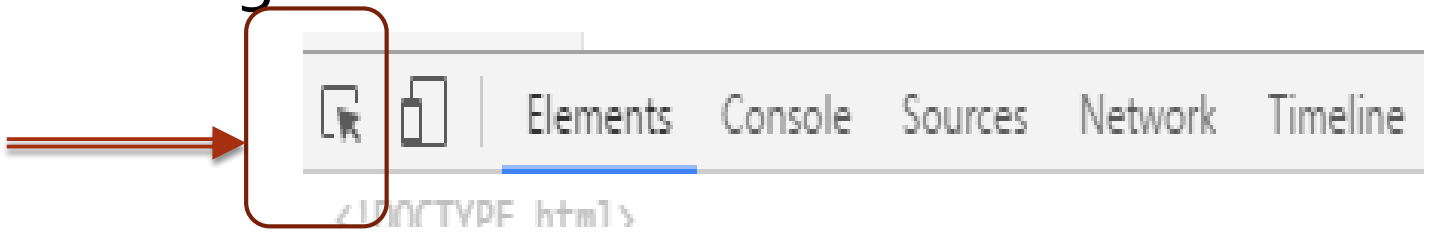
2. Apriamo lo strumento sviluppatori F12

3. Ispezioniamo gli elementi di interesse

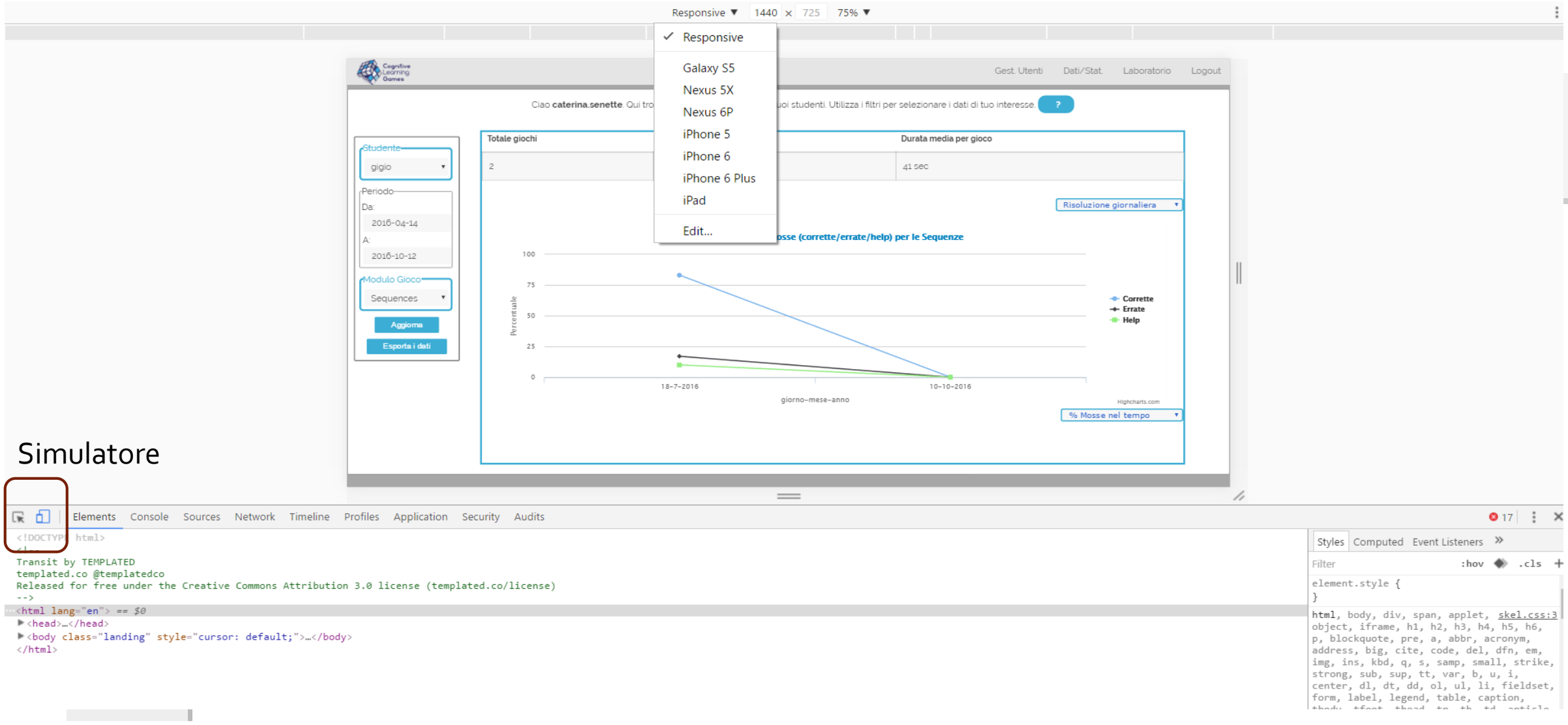
Strategie di SVILUPPO

Strategie di SVILUPPO

- **Network:** mostra i tempi di caricamento di ciascuna risorsa presente nella pagina (fogli di stile, file javascript, immagini, …)
- **Resources:** visualizza i contenuti di un documento (immagini, font, script, figli di stile) e i dati di alcune API come: WEB SQL, Local Storage, Session Storage, Cookies e Application Cache
- **Console:** visualizza eventuali errori o avvertimenti relativi a javascript
- **Strumento di ispezione** del singolo elemento



Strategie di SVILUPPO



Simulatore

Capitolo 3

Capire nella pratica quali elementi rendono una APP accessibile e usabile nei diversi contesti di disabilità

SYLLABUS

- Grafica ed elementi di interfaccia
- Interazione con l'utente
- Adattabilità/Personalizzazione

Usabilità

La norma ISO 9241-11:1998, poi aggiornata dalla ISO 9241-210:2010, la definisce come:

"Il grado in cui un prodotto può essere usato da **particolari utenti** per raggiungere **certi obiettivi** con efficacia, efficienza, soddisfazione in **uno specifico contesto d'uso**"

Obiettivi delle WEB USABILITY

- Presentare l'informazione all'utente in modo chiaro e conciso (NO termini tecnici o specialistici)
- Semplificare la struttura del compito
- Offrire all'utente le scelte corrette, in una maniera che risulti ovvia
- Organizzare ogni pagina in modo che l'utente riconosca la posizione e le azioni da compiere
- Eliminare ogni ambiguità relativa alle conseguenze di un'azione (es. fare clic su cancella/rimuovi/compra)
- Mettere la cosa più importante nella posizione giusta della pagina web o dell'applicazione web
- Fare in modo che l'utente abbia un rapido feedback (informazione di ritorno) ad ogni azione compiuta
- Rendere la grafica accattivante ed interessante dal punto di vista visivo attraverso l'uso di diagrammi, tabelle, sezioni informative
- Ridurre gli sforzi cognitivi dell'utente

OCCORRE QUINDI TENER CONTO:

- **Del compito che l'utente deve svolgere** (la navigazione delle pagine di un sito)
- **Dell'utente che lo svolge** (target cui si rivolge il sito)
- **Dell'ambiente d'uso** (interfaccia, hardware, browser, velocità di connessione)

I requisiti BASE che, nella pratica, permettono di realizzare un sito (APP) usabile sono i seguenti:

- navigabilità
- tempi di attesa ridotti
- completezza dei contenuti
- comprensibilità delle informazioni
- efficacia comunicativa
- attrattiva grafica

ACCESSIBILITÁ / USABILITÁ NEL NOSTRO CASO

Le applicazioni che rappresentano i nostri casi d'uso hanno l'obiettivo di soddisfare 2 macro tipologie di UTENTE:

c'è sempre un TUTOR (spesso un genitore) e uno STUDENTE

quindi lo studio

(i) degli elementi grafici
(ii)dell'interazione utente
(iii)dell'adattabilità/personalizzazione

va fatto in relazione a queste 2 macro tipologie con risultati (conclusioni) non sempre identici

PRIMO CASO DI STUDIO: CLG

Cognitive
Learning
Games

Profilo TUTOR

CASO CLG

Qual è il profilo del TUTOR?

- Ha familiarità con applicazioni SW ma non è particolarmente skillato
- Conosce le tipologie di esercizio che il SW offre ma le ha proposte, ai suoi studenti, quasi sempre in maniera tradizionale
- Ha necessità di un ambiente SW facile da usare che non rappresenti una distrazione mentre prende in carico lo studente con disagio cognitivo
- Vuole un ambiente che tenga traccia del lavoro svolto dallo studente
- Vuole poter controllare gli esercizi che lo studente fa (programmazione percorso training) ma auspica che lo studente lavori da solo

Grafica ed elementi di interfaccia-TUTOR

CASO CLG - Interfaccia del TUTOR

1. Si usano i CSS che definiscano ogni elemento HTML sempre allo stesso modo (es: BOOTSTRAP)

   - le select ,check box, i link, le caselle di testo ecc. sono sempre definite allo stesso modo in tutte le pagine dell'applicazione ove compaiono
   - colori tenui in tutta l'applicazione con tonalità differenti per distinguere funzionalità (o priorità) differenti
   - font coerenti
2. Presenza di scritte descrittive delle funzionalità offerte dalle varie pagine
3. Linguaggio informale dove possibile
4. Presenza di pagine di help chiare e brevi nei punti più delicati dell'applicazione

CASO CLG - Interfaccia del TUTOR

FOCUS SU GESTIONE UTENTI

1. Messa in evidenza dei dati correnti diversificati dal contesto
2. Stringhe di testo semplici e non ambigue
3. Menù di navigazione chiari e che siano in grado di agevolare l'orientamento dell'utente
4. Feedback continui (vedi inserimento nuovo utente che compare subito nella lista)
5. Reversibilità delle azioni (Tasto Annulla)

FOCUS SU LABORATORIO

1. Fornire le funzionalità in modo graduale e guidato esempio: creazione di 1 nuovo gioco partendo dal PRIMO STEP: nome del gioco…
2. Anticipare le potenziali azioni dell'utente per "servirle" in maniera intuitiva e semplificata
3. Visualizzazione immediata delle risorse disponibili
4. Presenza di filtri per la ricerca facilitata

CASO CLG - Interfaccia del TUTOR

FOCUS SU STATISTICHE

1. Offrire filtri di ricerca coerenti
2. Evidenziare gli elementi che richiedono input utente usando la grafica
3. Semplificare il numero di azioni dell'utente per trovare i dati che cerca
4. Offrire feedback qualora i dati non siano presenti

RESPONSIVENESS

CLG è un'applicazione che come requisito ha la possibilità di essere usata da dispositivi di diverse dimensioni nello specifico delle interfacce destinate al tutor questo ha qualche limite, per esempio difficilmente userò la sezione di statistiche su un cellulare quindi la sua responsiveness in questa sezione non è un requisito stringente

Background: Responsive Design

Soluzione 1: Uso del CSS3

Basta scrivere un solo foglio di stile dove, nella prima parte di codice inserisco gli elementi in comune e poi occorre suddividere il resto del codice per tipologia di display.

In CSS3 il richiamo alla tipologia del display lo si fa semplicemente con la dichiarazione che conosciamo:

```
@media only screen and (min-width: 768px) and (max-width: 959px) {...}
```

Soluzione 2 (necessaria quando la compatibilità con CSS3 non è garantita): **Uso di Jquery**

Ovvero si usa jQuery per rilevare la risoluzione del display, poi si scrive in un file (es:*screen.css*) tutti gli elementi che accomunano le versioni del sito. Mentre in fogli di stile separati (*small.css*, *large.css*, ...) si scriveranno le caratteristiche relative ad ogni singolo dispositivo.

A questo punto un'altra piccola libreria (tipo Adapt.js) si incarica di selezionare il file appropriato per lo schermo X.

Interazione Utente (profilo TUTOR)

CASO CLG - Interfaccia del TUTOR

L'interazione utente va sempre studiata in relazione agli elementi di interfaccia. Quindi una interfaccia ben disegnata rende l'interazione utente più semplice ed efficace perché:

- L'utente sa sempre dove si trova durante la navigazione
- L'utente sa cosa l'applicazione può offrire e sa come ottenerlo
- Ha feedback istantanei che lo informano del successo/insuccesso della sua azione
- Gestisce i suoi studenti e le risorse (immagini/giochi) senza necessità di competenza specifica perché ogni funzionalità offerta è semplificata
- Può usare pagine di Help in caso di dubbi su una certa funzionalità

Profilo STUDENTE

CASO CLG

Qual è il profilo dello STUDENTE?

- È un soggetto con disabilità cognitive lieve-moderata di età diverse
- Conosce le tipologie di esercizio che il SW offre ma solitamente svolge gli esercizi in maniera tradizionale
- Non ha necessariamente familiarità con l'uso del PC /dispositivi elettronici in genere
- Ha bisogno di un ambiente guidato eventualmente multimodale
- Ha una bassa capacità di concentrazione (e scarsa capacità di attesa) quindi deve essere opportunamente stimolato
- Necessita di contenuti personalizzati sui suoi interessi/preferenze
- Ha bisogno di un ambiente SICURO (NO navigazione WEB) che ne permetta l'uso in autonomia

Grafica ed elementi di interfaccia-STUDENTE

CASO CLG - Interfaccia dello STUDENTE

1. Pagine semplici e minimali che non contengano annidamenti (lo studente deve fare il numero minimo di azioni) per iniziare a giocare
2. Presenza di scritte descrittive delle funzionalità offerte dalle pagine soprattutto quelle di gioco
3. Ove possibile e richiesto le scritte devono essere sostituite da icone intuitive e/o prompt vocali
4. Linguaggio informale dove possibile
5. Attrattività grafica ottenuta con immagini e uso del colore per motivare il soggetto a lavorare (attenzione a eventuali iper-sensibilità)
6. Elementi della pagina sempre COERENTI (se la X chiude il gioco deve essere sempre usata con quel significato)
7. Pagina sicura, in modalità full screen per eliminare la possibilità che lo studente navighi nel web da solo

Interazione Utente (caso dello STUDENTE)

CASO CLG - Interfaccia dello STUDENTE

Come per tutte le applicazioni di tipo RIA un vincolo determinante è la **velocità di risposta dell'APPLICAZIONE** e questo è abbastanza facile garantirlo in alcune pagine dell'applicazione, diventa più difficile in altre, prima fra tutte la pagina di gioco

Alcuni soggetti con disabilità cognitiva (es Sindrome di Down) mostrano capacità di attesa minima (come accade tipicamente nei bambini), la lentezza nella risposta li rende facilmente frustrati

L'ostacolo maggiore è rappresentato da:

1. Caricamento immagini
2. Animazione delle stesse immagini

Adattabilità e Personalizzazione

CASO CLG

Esistono in rete tantissime applicazioni che propongono giochi cognitivi simili a quelli proposti da CLG.
Se esse non sono attentamente disegnate un ragazzo con sindrome di Down difficilmente le userà perché magari:

- I compiti sono percepiti troppo facili
- Oppure i compiti sono percepiti troppo difficili
- Non riescono a catturare la loro attenzione
- La «ricchezza» nella forma disorienta il soggetto

} **NON SONO ACCESSIBILI**

CLG è stata pensata per dare risposte a questo tipo di problematica e lo fa grazie alle caratteristiche seguenti

CASO CLG

CLG è stata pensata per dare risposte a questo tipo di problematica e lo fa grazie alle caratteristiche seguenti

1. Esercizi a differenti gradi di difficoltà
2. Personalizzazione dei contenuti (si possono creare tante istanze di gioco usando risorse personali)
3. Principio ERROR LESS
4. Rinforzo positivo in caso di successo
5. Uso della CAA tramite immagini
6. Semplicità nell'accesso al gioco

Grazie per l'attenzione !!!

Ing. CATERINA SENETTE
caterina.senette@iit.cnr.it