



Università degli studi di Milano Bicocca

**Dipartimento di informatica,  
sistemistica e comunicazione**

**Corso di laurea in informatica**

---

## **Sviluppo e-commerce: avifauna.fem2ambiente.com**

---

**Relatore:** Micucci Daniela

**Correlatore:** Ferri Emanuele

**Relazione della prova finale di**

Mattia Curatitoli

735722

**Anno accademico 2014-2015**

# Indice

<b>1. Introduzione</b>	<b>3</b>
<b>2. FEM2-Ambiente</b>	<b>4</b>
2.1. La storia di FEM2-Ambiente . . . . .	4
2.2. La richiesta . . . . .	5
<b>3. Sito in WordPress</b>	<b>7</b>
3.1. Caratteristiche di WordPress . . . . .	7
3.2. WordPress per avifauna.fem2ambiente.com . . . . .	7
<b>4. Sviluppo Piattaforma Web</b>	<b>10</b>
4.1. Lato Server . . . . .	10
4.1.1. Django . . . . .	10
4.1.2. Configurazione . . . . .	11
4.1.3. Creazione . . . . .	11
4.2. Lato Client . . . . .	17
4.2.1. HTML . . . . .	17
4.2.2. CSS . . . . .	18
4.2.3. JavaScript . . . . .	18
4.2.4. Bootstrap . . . . .	18
4.2.5. Vista Cliente . . . . .	20
4.2.6. Vista Admin . . . . .	25
<b>5. Conclusioni</b>	<b>26</b>
<b>A. Codice Sorgente</b>	<b>27</b>
<b>B. Guida Admin</b>	<b>30</b>

# 1. Introduzione

**FEM2-Ambiente Srl** è uno spin-off del Dipartimento di Biotecnologie e Bioscienze dell'Università degli Studi di Milano Bicocca, nato con l'intenzione di supportare i consumatori nelle scelte dei prodotti e servizi biologici. Dopo la sua nascita nel 2010, si è sviluppando ed è cresciuto sempre più grazie al supporto fornito dall'Università per la ricerca e grazie al suo impatto sul mercato con prodotti innovativi e di facile uso.

FEM2-Ambiente dispone di moderni laboratori ospitati presso l'Università, nei quali vengono sviluppati e testati i nuovi prodotti, e durante la crescita si sono concentrati anche su prodotti mirati all'avifauna. In un primo momento ha cominciato ad offrire servizi di analisi su uccelli e col passare del tempo il numero di analisi possibili è aumentato ed è ancora in crescita. È stato necessario quindi evolversi da un primo approccio col cliente attraverso fogli di excel e costruire una piattaforma in grado di gestire gli ordini e il flusso cliente.

Questo sviluppo è l'argomento della seguente relazione ed è stato l'ambito del lavoro di stage.

È stato necessario l'utilizzo di diverse tecnologie in base alle richieste: dalla scelta di un CMS come *WordPress* per la parte del sito più informativa ed espositiva, in grado di essere facilmente usabile ad aggiornabile da persone non tecniche del settore informatico, alla scelta di framework per una piattaforma autonoma ma connessa al sito principale che permettessero la gestione completa di un flusso ordini personalizzato come quello del Portale Avifauna.

Nel capitolo 2 sarà descritta meglio l'azienda FEM2-Ambiente Srl e la sua storia, fino ad arrivare alla richiesta dello sviluppo della piattaforma web, nel capitolo 3 verrà descritta la scelta del CMS WordPress e la sua installazione, mentre il capitolo 4 è dedicato al vero e proprio sviluppo del Portale attraverso la descrizione di tutti i tool e framework utilizzati.

Per approfondire alcune porzioni di codice significative è stata scritta l'Appendice A con attenzione ad alcuni particolari funzioni fornite dal framework Django.

Invece per descrivere meglio i compiti lato admin e il flusso cliente sono state scritte rispettivamente le appendici B e ??.

## 2. FEM2-Ambiente

In questo capitolo viene descritta l'azienda **FEM2-Ambiente Srl** in modo da contestualizzare i bisogni che hanno portato allo sviluppo della piattaforma web dedicata.

In particolare nelle sezioni 2.1 e 2.2 sono illustrate prima nascita ed espansione della spin-off, poi bisogni, richieste e necessità legate alla piattaforma sviluppata.

### 2.1. La storia di FEM2-Ambiente

**FEM2-Ambiente Srl** è uno spin-off del Dipartimento di Biotecnologie e Bioscienze dell'Università degli Studi di Milano Bicocca, nato con l'intenzione di creare prodotti e servizi per il largo pubblico finalizzati alla conoscenza e tutela della biodiversità. La mission è supportare i consumatori nelle scelte, rendendoli consapevoli sulla qualità delle risorse ambientali, e in questo modo fornendo gli strumenti necessari a migliorare il loro stile di vita, tutelando l'ambiente [1].



**FEM<sup>2</sup> AMBIENTE**  
SOCIETÀ SPIN-OFF  
DELL'UNIVERSITÀ  
DEGLI STUDI DI  
MILANO-BICOCCA

Ad inizio 2007 è nato **ZooPlantLab** dall'incontro di Massimo Labra e Maurizio Casiraghi, due ricercatori del Dipartimento di Biotecnologie e Bioscienze dell'Università degli Studi di Milano Bicocca che si occupano rispettivamente di tematiche botaniche e zoologiche. Lo ZooPlantLab è un laboratorio di ricerca di zoologia, botanica e microbiologia che coniuga ricerca di base e applicata con progetti che prevedono l'utilizzo di un approccio molecolare [2].

Nel gennaio 2010 grazie al contributo e i risultati della ricerca di ZooPlantLab viene fondato FEM2-Ambiente Srl dai quattro soci fondatori: Dott. De Mattia Fabrizio, Dott. Ferri Emanuele, Dott. Labra Massimo, Dott. Casiraghi Maurizio.

Food, Environment & ManageMent (FEM2): alimentazione, ambiente e gestione razionale sono alcuni degli aspetti ai quali FEM2-Ambiente dedica particolare attenzione, con l'ottica di fornire informazioni e strumenti per un utilizzo più con-

sapevole delle risorse, in sintonia con il pianeta, utilizzando tecnologie e conoscenze derivanti dalla ricerca scientifica.

FEM2-Ambiente dispone di moderni laboratori ospitati presso l'Università, nei quali vengono sviluppati e testati i nuovi prodotti, eseguite analisi su matrici ambientali (es: acqua, aria o alimenti), si svolgono analisi del DNA e vengono messe a punto metodiche innovative di caratterizzazione molecolare. Grazie ad essi oggi FEM2-Ambiente, pur mantenendo le sue solide radici universitarie ed investendo nella ricerca, si è affermata anche come società commerciale e propone al mercato nazionale ed internazionale prodotti e servizi all'avanguardia nei settori dell'ambiente, del food e della diagnostica molecolare avanzata.

Negli ultimi anni è diventato leader di mercato nella *diagnostica molecolare di avifauna* tramite PCR (analisi del DNA), ed è nata la necessità di sviluppare una piattaforma adatta per gestire tutte le fasi di analisi e vendita dei servizi.

## 2.2. La richiesta

La crescita di FEM2-Ambiente sul mercato ha portato alla creazione del sito dedicato [www.fem2ambiente.com](http://www.fem2ambiente.com) basato su *Joomla!*, un *CMS* (content management system) molto diffuso, utile per la gestione dei contenuti del sito web senza la necessità di avere conoscenze tecniche [3]. Ad esso è stata aggiunta l'estensione *VirtueMart*, una soluzione open-source per la gestione dell'e-commerce che ad oggi permette l'acquisto di prodotti per l'analisi di acqua e aria rivolti a privati, condomini, imprese e per l'educazione nelle scuole, oltre ad eco-prodotti per la casa e per la cura degli animali [4].

Quando FEM2-Ambiente ha cominciato l'analisi su avifauna l'utilizzo di fogli elettronici Excel è sembrata la soluzione migliore, ma col crescere della clientela e del numero di ordini si è dovuto pensare ad una alternativa.

È nata così la richiesta di una piattaforma online dedicata in grado di offrire servizi di *sessaggio*, *diagnosi patologie* e *Dna barcoding*. In particolare in grado di gestire la vendita specifica delle analisi e servizi offerti (non realizzabile attraverso un semplice tool e-commerce già esistente), e il controllo del flusso ordini integrato con i procedimenti in laboratorio.

Sono stati sviluppati di conseguenza

- un sito dedicato all'esposizione dei servizi offerti, il *Portale della Diagnostica Molecolare dedicato all'Avifauna* (capitolo 3)

- una piattaforma lato server per il controllo completo del flusso ordini (capitolo 4.1)
- un interfaccia per il flusso cliente durante la creazione e monitoraggio degli ordini e un pannello admin per la gestione completa di ogni parte della piattaforma (capitolo 4.2)

La mole di lavoro da compiere ha creato la necessita di dividere i compiti, così io mi sono occupato dello sviluppo lato client/front-end dell'applicativo e del controllo del CMS WordPress; il lato server é stato invece compito di due colleghi.

Nei seguenti capitoli saranno descritti ed analizzati tutti i passi fino alla messa online del sito con particolare attenzione al lato front-end. Nel dettaglio il capitolo 3 descriverà le azioni per impostare correttamente il CMS, la sezione 4.1 fornirà una panoramica sul lato server, mentre il lato client sarà descritto più dettagliatamente nella sezione 4.2.

## 3. Sito in WordPress

L'analisi dei requisiti ha evidenziato la necessità di un sito web in cui esporre e descrivere al cliente i servizi offerti, aggiornato e popolato da un componente del team di FEM2-Ambiente senza per forza conoscenze e capacità tecniche informatiche; si è scelto quindi WordPress, un CMS molto diffuso.

WordPress è una piattaforma software di content management system (CMS) ovvero un programma installato sul server che consente la creazione, gestione, distribuzione e manutenzione di un sito Internet [5]. È un progetto open-source creato da Matt Mullenweg e distribuito con la licenza GNU General Public License; è sviluppato in PHP con appoggio a MySQL come gestore di database.

WordPress permette il download gratuito di tutti i suoi componenti dal sito [www.wordpress.org](http://www.wordpress.org) per poterli installare sulla propria macchina. Esiste anche un servizio (a pagamento in base alle richieste) chiamato *WordPress.com* che permette di costruire rapidamente il proprio sito web o blog basato su WordPress senza la necessità di possedere un server o competenze tecniche specifiche.

### 3.1. Caratteristiche di WordPress

WordPress permette di estendere le proprie funzionalità con l'ausilio di opportuni plugin, ovvero moduli che aggiungono nuove caratteristiche ed elementi all'applicativo. I plugin possono essere gratuiti o a pagamento e possono fare molte cose di tutto, dal potenziare l'editor integrato di WordPress all'inserire slideshow nelle pagine, e molto altro ancora. Come i plugin si possono trovare anche temi, estensioni che permettono di personalizzare l'aspetto del sito modificando sfondi, impaginazione, font, etc.

### 3.2. WordPress per [avifauna.fem2ambiente.com](http://avifauna.fem2ambiente.com)

Per realizzare il *Portale della Diagnostica Molecolare dedicato all'Avifauna*, dopo aver scelto il sottodominio [www.avifauna.fem2ambiente.com](http://www.avifauna.fem2ambiente.com), si è prima di tutto

installato e configurato WordPress.

Per farlo é stato necessario scaricare l'ultima versione dal sito [www.wordpress.org](http://www.wordpress.org) (ad oggi, Ottobre 2015, l'ultima versione é la 4.3.1) e seguire le istruzioni nel file *readme* [6], in particolare:

- eseguire le opportune modifiche al file `wp-config.php` in un editor di testo;
- creare un database dedicato utilizzando MySQL;
- connettersi al server e caricare tutti i file relativi l'installazione di WordPress nella cartella scelta (`/home`);
- configurare in modo appropriato visitando la pagina

<http://avifauna.fem2mambiente.com/home/wp-admin/install.php>

Una volta terminata l'installazione si é potuto procedere con l'installazione degli appropriati plugin, temi e estensioni.

La scelta del tema é ricaduta su *Everest* di YOOftheme (Versione: 1.0.11). YOOftheme é una azienda tedesca che produce componenti per CMS [7]; i loro prodotti più importanti, oltre a una ventina di temi e template diversi rispettivamente per WordPress e Joomla!, sono *Wrap Framework* [8] e *Uikit* [9], due architetture software di supporto per la creazione e personalizzazione dei componenti aggiuntivi ai più famosi CMS.

Il tema Everest é stato costruito utilizzando Wrap Framework e mette a disposizione dell'utilizzatore sette stili di layout differenti, personalizzazioni nella costruzione del layout sfruttando tutte le potenzialità di WordPress e un pacchetto di plugin chiamato *Widgetkit* per l'inserimento rapido di Slideshow, gallerie di immagini, mappe. Per tutte queste caratteristiche é stato scelto, acquistato ed installato come tema per il sito.

Per ricoprire tutti i bisogni organizzativi di un sito commerciale come il Portale Avifauna é stato necessario installare anche plugin come *Polylang* [10] per il supporto multilingue al sito e *My Calendar* [11] per la gestione degli eventi.

Dopo aver creato popolato il sito con i contenuti, divisi in base alle pagine e sezioni dedicate, il risultato ottenuto é visibile nell'immagine 3.1 e al seguente link [www.avifauna.fem2mambiente.com/home](http://www.avifauna.fem2mambiente.com/home).

Cliccando sul tasto **Ordini** si può accedere alla piattaforma personalizzata, descritta nei seguenti capitoli.



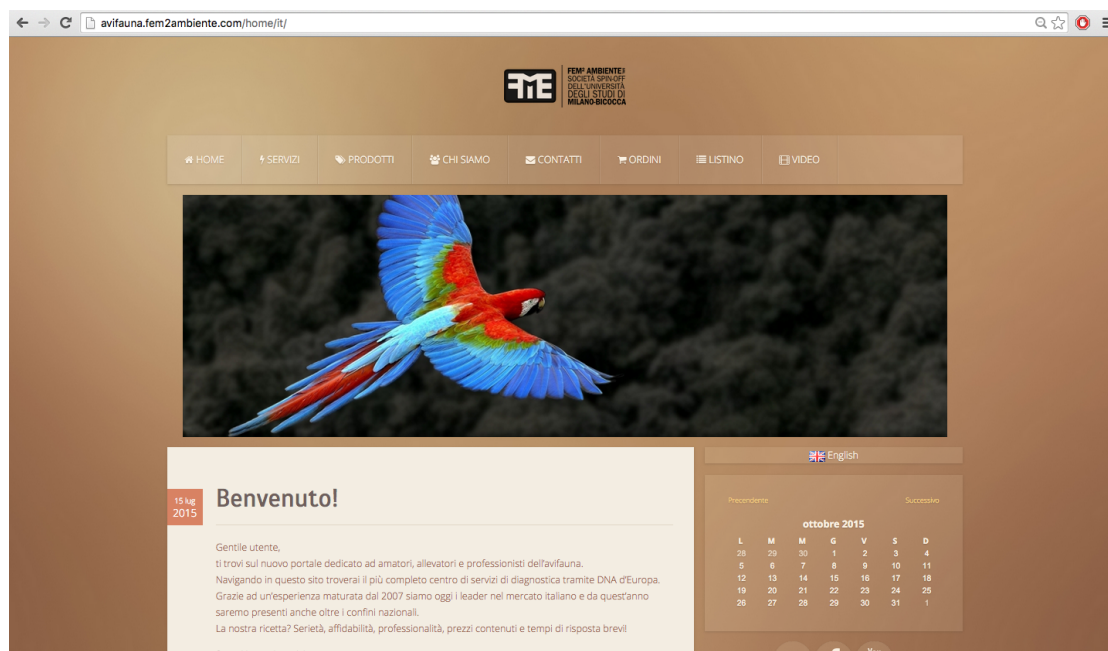
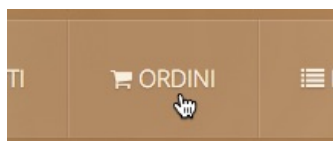


Figura 3.1.: homepage del Portale per la Diagnostica Molecolare Avifauna



## 4. Sviluppo Piattaforma Web

In questo capitolo é descritta la costruzione della piattaforma acquisti associata al sito divulgativo creato con WordPress. Lo sviluppo può essere concettualmente diviso in due parti: il *lato server* e il *lato client*. Esse si concentrano rispettivamente sulla creazione di una base su cui interagire per tenere traccia di tutte le azioni compiute nel flusso di acquisto e analisi e sull'interfaccia con l'utente finale, che può essere il cliente oppure un addetto di FEM2-Ambiente.

### 4.1. Lato Server

Per lo sviluppo della piattaforma web le tecnologie utilizzate sono state: *Django* come web framework e *MySQL* per il database.

#### 4.1.1. Django

*Django* è un web framework open source per lo sviluppo di applicazioni web, scritto in linguaggio *Python*; il progetto è sviluppato dalla "Django Software Foundation" (DSF), un'organizzazione indipendente senza scopo di lucro [12]. É stato inizialmente concepito per gestire diversi siti di notizie, ed in seguito distribuito con una licenza BSD (Berkeley Software Distribution) a luglio 2005.

La scelta di Django é ricaduta grazie alle molte proprietà: dall'astrazione del database relazionale ad oggetti, alla possibilità di installare funzionalità attraverso plugin, dalla robusta API per la gestione del database, al sistema di "view generiche" che evitano la stesura di codice ripetitivo per determinati casi comuni e soprattutto il sistema di template e gestore di URL basate su espressioni regolari. Django offre inoltre un efficace supporto per localizzazione, incluse traduzioni dell'interfaccia amministrativa in molte lingue.

### 4.1.2. Configurazione

Il primo passo é stato l'installazione delle componenti di Django, attraverso la creazione del progetto (nome di esempio `mysite`), con il comando a terminale:

```
$ django-admin startproject mysite
```

così da ottenere la seguente configurazione di file:

```
mysite/  
  manage.py  
  mysite/  
    __init__.py  
    settings.py  
    urls.py  
    wsgi.py
```

Abbiamo quindi impostato nel file `settings.py` il database scelto, le lingue del sistema, il percorso dei file statici, dei media e le `INSTALLED_APPS`.

Le `INSTALLED_APPS` sono una sorta di librerie usate per l'aggiunta di componenti al progetto costruito; le più importanti sono:

- `django.contrib.admin` - il creatore automatico del pannello admin
- `django.contrib.auth` - il sistema di autenticazione
- `django.contrib.sessions` - il framework per il controllo delle sessioni
- `django.contrib.messages` - il framework di controllo per i messaggi
- `django.contrib.staticfiles` - il gestore dei "file statici"

### 4.1.3. Creazione

Durante tutta la fase di sviluppo é stato necessario avviare il *development server* attraverso il comando da terminale:

```
$ python manage.py runserver
```

per simulare il comportamento del server in modo da generare il sito all'indirizzo `http://127.0.0.1:8000/`.

Una componente fondamentale é rappresentata dai *modelli*, strutture associabili concettualmente alle classi in Java; in funzione alle richieste avanzate da FEM2-Ambiente il diagramma in figura 4.1 rappresenta come sono stati configurati i modelli con i relativi attributi.

## 404 - File or directory not found.

The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Figura 4.1.: configurazione modelli

nei seguenti paragrafi sono descritti i principali modelli.

### clienti.py

Il cliente é un componente delicato ed importante del sistema.

L'attributo `tipo` é necessario in quanto per FEM2-Ambiente il cliente può essere differenziato in quattro tipi: Amatore, Allevatore, Veterinario e Negozio. Esso é identificato univocamente dall'indirizzo `email` inserito al momento della registrazione e ha chiaramente un attributo `ragione_sociale` per indicare nome e cognome per un privato, oppure ragione sociale in caso contrario. Ogni cliente ha altri numerosi attributi per ogni dato personale relativo all'indirizzo (necessario per la spedizione degli attestati generati al termine delle analisi), contatti telefonici e `lingua_preferita` per tradurre il sistema correttamente.

Ogni cliente può acquistare pacchetti di *Crediti FEM*, cioè una somma di denaro pronta per gli acquisti pagata anticipatamente, in modo da non dover effettuare il pagamento al termine di ogni ordine, é quindi necessario indicare la quantità di crediti posseduta da ogni cliente in un apposito attributo.

Infine ogni cliente può essere iscritto ad una associazione convenzionata all'azienda FEM2-Ambiente e deve poter inserire il proprio numero di tessera per accedere agli sconti relativi; per farlo si deve dare uno sguardo ai modelli `associazioni.py` e `prezzi.py`.

### prezzi.py

Ogni `SchemaPrezzi` é identificato dal `nome` e può essere associato a nessuno, uno o tanti clienti. Esso definisce i costi fissi delle commissioni per ogni metodo di pagamento scelto, il prezzo degli attestati e il prezzo di ogni analisi, che tendenzialmente può variare tra uno `SchemaPrezzi` e l'altro. In particolare abbiamo deciso di differenziare gli `SchemaPrezzi` secondo una caratteristica principale: se schema *Pacchetti* o *Convenzioni*.

## 404 - File or directory not found.

The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Figura 4.2.: flusso ordine

Uno SchemaPrezzi del tipo *Pacchetti* é caratteristico di un cliente standard, che può usufruire di sconti vincolati a quantità, ad esempio con l'acquisto di un analisi APV associato ad un analisi SMAP riduce il costo di entrambe. Uno SchemaPrezzi del tipo *Convenzioni* invece é adatto per i clienti che risultano iscritti ad una associazione che ha attiva una convenzione con FEM2-Ambiente; questo tipo di SchemaPrezzi modifica il prezzo di tutte o alcune analisi anche in relazione alle specie del campione scelta.

### associazioni.py

Una Associazione é caratterizzata da un nome e da uno schema prezzi associato. `IscrizioneAssociazione` indica la correlazione tra un cliente, indicato attraverso nome e numero di tessera, e una associazione.

### ordini.py

L' `Ordine` é il componente più delicato e complesso del sistema a causa del suo flusso rappresentato in figura 4.2.

Esso é caratterizzato da un numero, dal `cliente` che l'ha creato e dall'attributo `stato` che indica in quale posizione del flusso si trova. Altri attributi interessanti sono:

- `metodo_pagamento` indica il metodo di pagamento scelto
- `totale_servizi` indica il costo delle analisi richieste con aggiunte le eventuali spese di spedizione per gli attestati cartacei
- `crediti_consumati` indica la quantità di crediti FEM utilizzati per pagare l'ordine
- `servizi_da_pagare` indica il `totale_servizi` da cui sono stati sottratti i `crediti_consumati`



Figura 4.3.: Classificazione biologica secondo i principali ranghi tassonomici

- `ammontare` indica il `totale_servizi` sommato all'eventuale costo di commissione previsto dal metodo di pagamento

## specie.py

É necessario prima di tutto spiegare che la tassonomia (dal greco *taxis* "ordinamento", e *nomos* "norma" o "regola") é definita in generale come la disciplina della classificazione. Abitualmente si impiega il termine per indicare la tassonomia biologica, ossia la disciplina scientifica che si occupa di attribuire un nome agli organismi viventi e di classificarli. La gerarchia di classificazione biologica secondo gli otto principali ranghi tassonomici é descritta in figura 4.3 (le posizioni intermedie alla classifica non sono visualizzate).

Il Portale Avifauna si occupa per definizione di analisi su un complesso di uccelli che vivono in una determinata regione, ma non tutti gli esemplari richiedono lo stesso tipo di approccio o tecnica per effettuare la analisi. La differenziazione principale scelta é quindi in base a *specie* e sottospecie.

Il modello `specie.py` si occupa di creare la base per questa differenziazione. Ogni specie ha `ordine`, `famiglia` e `nome` oltre ad un `immagine` per descriverla; ogni sottospecie ha la `specie` di riferimento, il `nome` e i possibili `nomi_comuni` e `protonimi`.

Le sottospecie sono anche caratterizzate dalla proprietà `junior` ovvero considerate 'nuove' per FEM2-Ambiente che non ha ancora acquisito un numero soddisfacente di dati relativi alla sottospecie in questione, quindi non garantisce al 100% il corretto risultato delle analisi di tipo sessaggio.

Per facilitare il lavoro in laboratorio sono stati creati anche `metodo_di_estrazione`, `metodo_di_amplificazione`, `metodo_di_visualizzazione` utili ai tecnici per riconoscere più rapidamente come eseguire le operazioni di analisi.

## **campioni.py**

Analogamente ad un e-commerce tradizionale ogni ordine é una sorta di carrello che contiene uno o più oggetti all'interno; nel Portale Avifauna gli oggetti sono i **campioni**, ovvero gli esemplari su cui sono richieste le analisi da effettuare.

Ogni campione ha i seguenti attributi:

- **ordine** per indicare il numero di ordine di riferimento
- **identificativo** ovvero il nome dell'esemplare
- **specie** o un eventuale **altra\_specie** in caso in cui non sia presente nell'elenco fornito da FEM2-Ambiente
- **mutazione**, **proprietario** e **data\_di\_nascita** utili soprattutto per la generazione degli attestati
- **lab**, **problema\_campione**, **standard**, **voucher** utili per il lavoro in laboratorio

Il campo **identificativo** segue le regole fornite dalla *Federazione Ornicoltori Italiani* (F.O.I.), un ente che raggruppa tutti gli appassionati ornicoltori e gli allevatori di uccelli con lo scopo di promuovere lo studio, il miglioramento, lo sviluppo e la conservazione del patrimonio ornitologico [13]. La F.O.I. ha regolamentato che per poter partecipare alle Manifestazioni Ornitologiche occorre che gli uccelli abbiano alla propria zampa un anellino che riporta i dati dell'allevatore (mediante la sigla *R.N.A.*), l'anno di nascita del soggetto e un numero progressivo, attraverso il quale è possibile risalire ai genitori, contattando l'allevatore che avrà avuto cura di registrare i dati genealogici in appositi registri. Il campo **identificativo** indicherà quindi il nome dell'esemplare in caso di privati e **RNA+Anello** in caso di allevatori professionisti.

Per facilitare il lavoro in laboratorio sono stati introdotti gli attributi **lab** che indica il numero progressivo e univoco in laboratorio e **problema\_campione** per indicare eventuali caratteristiche del campione, che può essere normale, mancante, inadatto, rischioso o annullato. Altri attributi interessanti sono **voucher** che funge da 'etichetta' per evidenziare un campione che viene preso di riferimento per future analisi e soprattutto **standard**.

Una specie, come spiegato precedentemente in 4.1.3, può essere definita *junior* in caso in cui FEM2-Ambiente lo reputi necessario in quanto non ha ancora eseguito un numero minimo di analisi di tipo sessaggio sulla specie in questione per sentirsi fiduciosa da assicurare il successo delle analisi al 100%. Per questo motivo, al momento della creazione dell'ordine, se viene aggiunto un campione di specie *junior*,

si chiede di aggiungere eventuali campioni (se il cliente ne é in possesso) definiti *standard*, ovvero esemplari di cui il cliente é già a conoscenza del sesso (tipicamente i genitori) in modo da aumentare le possibilità di successo delle analisi, poichè in laboratorio i tecnici di FEM2-Ambiente avrebbero maggiori punti di riferimento.

Su ciascun **campione** possono essere eseguite una o più *analisi*.

## **analisi.py**

Ad oggi FEM2-Ambiente é in grado di offrire servizi di analisi di tipo *Sessaggio molecolare* (determinazione di genere maschio o femmina) e *identificazione di patologie*.

I primi sono suddivisi in:

- **SMAP, Sessaggio Molecolare di Avifauna da Piuma:** analisi del DNA a partire da piume per stabilire il sesso del soggetto
- **SMAU, Sessaggio Molecolare di Avifauna da Uovo:** analisi del DNA a partire da frammenti di uovo
- **SMAR, Sessaggio Molecolare di Avifauna Rapido:** analisi rapida del DNA a partire da piume

I secondi in:

- **APV-Avian Polioma Virus:** un agente patogeno virale diffuso in tutto il mondo, in grado di infettare un ampio spettro di uccelli; poichè gli adulti solitamente sono portatori asintomatici, sono i principali responsabili della persistenza, della trasmissione e della diffusione della malattia. FEM2-Ambiente esegue analisi di screening di APV attraverso PCR (Polymerase Chain Reaction) a partire da piume e/o da prelievo ematico.
- **BFDV Circovirus:** virus che attacca i tessuti di becco, piume e artigli causando progressive malformazioni fino alla necrosi. Le analisi vengono eseguite attraverso PCR a partire da piume.
- **Clamidia - Chlamydophila psittaci:** un batterio che si può trovare nel torrente circolatorio, nei tessuti, negli escrementi e nelle piume degli uccelli.

Le analisi sono quindi caratterizzate dal **tipo\_servizio** tra quelli elencati precedentemente e dal **campione** di riferimento. Hanno un campo dedicato all'**esito** (di tipo Maschio, Femmina o Fallito nel caso di analisi di sessaggio; Positivo, Negativo o Fallito in caso di identificazione di patogeno) e delle variabili booleane per indicare se é stato richiesto **attestato\_cartaceo** o **attestato\_digitale**.



## 4.2. Lato Client

La struttura alla base della piattaforma web per il Portale Avifauna può essere rappresentata attraverso la divisione dei modelli descritta in precedenza; di seguito è invece descritta l'interfaccia utente sia per il cliente finale che per i tecnici di FEM2-Ambiente che lavorano tutti i giorni con il sistema.

Per la costruzione del lato client le tecnologie utilizzate sono *HTML*, *CSS*, *JavaScript* e *Bootstrap*.

### 4.2.1. HTML

*HTML* (HyperText Markup Language) è il linguaggio di markup utilizzato per la formattazione e impaginazione di documenti ipertestuali disponibili nel World Wide Web sotto forma di pagine web. È un linguaggio di pubblico dominio, la cui sintassi è stabilita dal World Wide Web Consortium (W3C) [14]; esso è un linguaggio di formattazione che descrive le modalità di impaginazione o visualizzazione grafica (layout) del contenuto, testuale e non, di una pagina web attraverso tag di formattazione.

L'HTML è stato sviluppato verso la fine degli anni ottanta da Tim Berners-Lee al CERN di Ginevra insieme al protocollo HTTP dedicato al trasferimento di documenti in tale formato. Negli anni novanta il linguaggio ha avuto una forte diffusione in seguito ai primi utilizzi commerciali del web. Attualmente i documenti HTML sono in grado di incorporare molte tecnologie, che offrono la possibilità di aggiungere al documento ipertestuale controlli più sofisticati sulla resa grafica, interazioni dinamiche con l'utente, animazioni interattive e contenuti multimediali. Si tratta di linguaggi come CSS, JavaScript e jQuery.

Il componente principale della sintassi di questo linguaggio è l'elemento, inteso come struttura di base a cui è delegata la funzione di formattare i dati o indicare al browser delle informazioni; ogni elemento è racchiuso all'interno di marcature dette tag, costituite da una sequenza di caratteri racchiusa tra due parentesi angolari o uncinate (<>).

L'ultima versione è detta *HTML5*, pubblicata come W3C Recommendation da ottobre 2014. Le novità introdotte dall'HTML5 sono finalizzate soprattutto a migliorare il disaccoppiamento fra struttura, definita dal markup e contenuti di una pagina web.

### 4.2.2. CSS

*CSS* (Cascading Style Sheets) è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML. Le regole per comporre il CSS sono contenute in un insieme di direttive (Recommendations) emanate a partire dal 1996 dal W3C [15].

L'introduzione del CSS si è resa necessaria a partire dalla metà degli anni novanta per separare i contenuti dalla formattazione e permettere una programmazione più chiara e facile da utilizzare, sia per gli autori delle pagine HTML che per gli utenti. Ad oggi l'ultima versione è *CSS3*.

### 4.2.3. JavaScript

*JavaScript* è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client per la creazione, in siti ed applicazioni web, di effetti dinamici interattivi tramite funzioni di script. Fu originariamente sviluppato da Brendan Eich della Netscape Communications con il nome di Mocha e successivamente di LiveScript, ma in seguito è stato rinominato JavaScript; è stato standardizzato per la prima volta nella fine degli anni novanta con il nome *ECMAScript* e l'ultimo standard, di giugno 2015, è ECMA-262 Edition 6 [16].

Una delle caratteristiche principali di JavaScript è di essere un linguaggio interpretato; in JavaScript lato client, l'interprete è incluso nel browser che si sta utilizzando il quale, quando viene visitata una pagina web che contiene il codice di uno script JavaScript, porta in memoria primaria lo script e lo esegue. Le interfacce che consentono a JavaScript di rapportarsi con un browser sono chiamate DOM (Document Object Model).

Molti siti web usano la tecnologia JavaScript lato client per creare potenti applicazioni web dinamiche, tra cui il Portale Avifauna per garantire una completa attenzione al cliente durante le procedure di creazione dell'ordine.

### 4.2.4. Bootstrap

*Bootstrap* è un framework che raccoglie strumenti liberi per la creazione di siti e applicazioni per il web, tra cui modelli basati su HTML, CSS e JavaScript per il controllo di struttura, tipografia e interfaccia [17].

Bootstrap è stato sviluppato da Mark Otto e Jacob Thornton a Twitter come un framework per uniformare i vari componenti utilizzati fino a quel momento ed è stato rilasciato nell'agosto 2011 come progetto open source.

Bootstrap è compatibile con le ultime versioni di tutti i principali browser e dalla versione 2.0 supporta anche il responsive web design, quindi anche il supporto al 'mobile web' da dispositivi mobili come tablet e smartphone. Ad oggi l'ultima versione è la 3.3.5 di maggio 2015, ma è già in fase di sviluppo la 4.0 [18].

Associato a Bootstrap è stato utilizzato anche *Font Awesome*, un toolkit di font e icone basato su CSS, creato da Dave Gandy per l'utilizzo in Twitter Bootstrap [19] [20]. Esso permette una maggiore personalizzazione di icone e tipografia molto utile nella Piattaforma Avifauna per comunicare graficamente in modo immediato con il cliente.

## 404 - File or directory not found.

The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Figura 4.4.: flusso cliente

### 4.2.5. Vista Cliente

Il cliente finale può compiere poche azioni basilari nella piattaforma sviluppata rappresentate nella figura 4.4. Queste azioni però devono essere supportate da un adeguata interfaccia e per farlo sono stati utilizzati tutti gli strumenti descritti in precedenza.

Django, essendo un web framework, ha bisogno di una gestione dinamica dei file HTML, e l'approccio più comune è l'utilizzo dei *templates*. Un template contiene porzioni statiche del codice HTML desiderato e permette la riproduzione di alcune o intere porzioni di codice in altri file per evitare inutili duplicati.

Sono stati quindi introdotti alcuni comandi:

- per includere un file HTML all'interno di un altro

```
{% include "file_name" %}
```

- per indicare un blocco di contenuto specifico; ad esempio inserendo `jsexec` al posto di `'...'` si indica una porzione dedicata a script JavaScript, oppure scrivendo `title` si indica il titolo della pagina HTML

```
{% block ... %} {% endblock %}
```

- per descrivere come il file HTML richieda di essere una estensione di un altro file (molto utile per creare e modificare lo stile di base di tutti i file generati in uno unico, in questo progetto chiamato `base.html`).

```
{% extend "file_name" %}
```

Alcuni esempi di utilizzo sono mostrati nell'appendice A.

Con la creazione del file `base.html` si sono definiti file statici come CSS e JavaScript una sola volta, generando uno stile solido e centralizzato che permette però personalizzazioni.

Dalla figura 4.4 del flusso cliente si evince come il primo passo sia stato costruire una sorta di Homepage per introdurre i clienti non registrati al Portale (file chiamato `home.html`) e da qui, attraverso la barra di navigazione, dare la possibilità di effettuare il login o la registrazione.

## registrazione e pagina personale

La pagina di registrazione (`register.html`) ha richiesto l'utilizzo di codice JavaScript principalmente per il controllo dei dati inseriti nei campi del form; in particolare il controllo in browser della compilazione di tutti i campi obbligatori e della loro correttezza sintattica in modo da evitare inutili comunicazione con il server che porterebbero a una, seppur lieve, perdita di tempo.

I controlli sono dei semplici 'validatori' che assegnano `true` alla variabile `any_error` in modo da visualizzare un messaggio di errore al momento dell'invio del form. Per rendere ancora più esplicito il messaggio di errore é stato inserito in un *modale*, strumento introdotto da Bootstrap che consiste in un `<div>` al centro della viewport (ovvero la regione di pagina che viene visualizzata nel monitor) che monopolizza l'attenzione. All'interno di esso viene inserito il testo relativo all'errore. Ad esempio

```
if(any_error) {  
    modal  
    .body("Sono presenti alcuni errori")  
}
```

Una volta completato il processo di registrazione il cliente visualizza la propria pagina personale (figura 4.5) in cui può vedere la propria cronologia di ordini, modificare i dati personali, acquistare pacchetti crediti FEM, ma soprattutto creare un nuovo ordine.

## nuovo ordine

La pagina di creazione dell'ordine é divisa in tre blocchi come sono le sue fasi: *carrello*, *riepilogo*, *pagamento*.

La prima schermata del carrello é divisa in due parti (figura 4.6). In alto si trova la tabella che funge da carrello vero e proprio raccogliendo tutti i campioni inseriti dal cliente e mostrando i relativi dati associati come identificativo, specie, analisi richieste, costo, etc. In basso si trova il form per l'inserimento di dei soggetti.

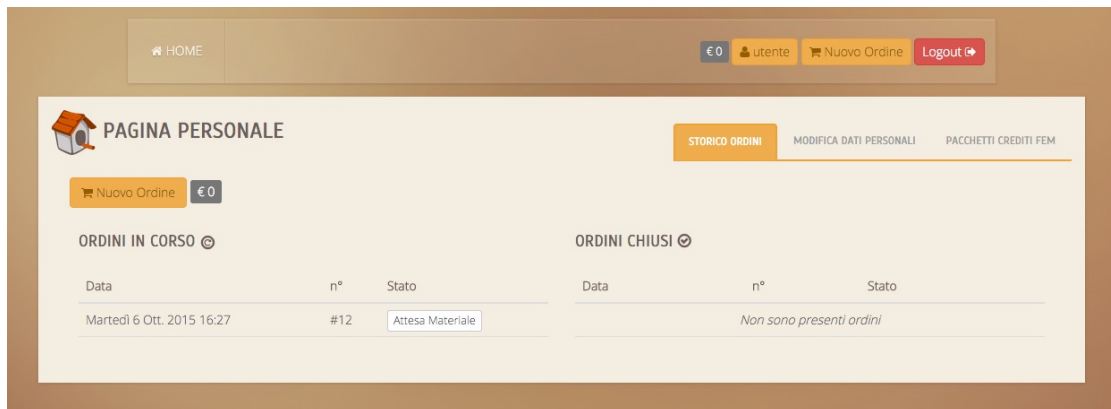


Figura 4.5.: pagina personale

Come nella pagina di registrazione viene utilizzato del codice JavaScript per la validazione dei campi, inoltre il campo **specie** funge da elenco in cui cercare la specie/sottospecie del soggetto.

Una volta selezionata la specie, a destra verrà popolato un `<div>` (identificato dalla classe `taxonomy`) contenente l'immagine e altre informazioni come i nome comuni come descritto nel seguente estratto di codice; se non è stata correttamente selezionata la specie tra quelle esistenti il `<div> taxonomy` non deve comparire. Per farlo è stata creata una classe apposta in CSS (`dno`) che va a impostare `display:none`, cioè oggetto non visibile. Di seguito se la specie è stata selezionata correttamente si popolano gli appositi `<div>` innestati (identificati dagli `id="nome_specie"` e così via) e l'immagine; quest'ultima potrebbe non essere presente per alcune specie, in questi casi si caricherà un'immagine preimpostata per indicarlo.

```
if($(this).val() == "-1") {
    $(".taxonomy").addClass("dno");
} else {
    if(data_Specie[$(this).val()].immagine == "") {
        $(".image-exist").addClass("dno");
        $(".image-not-exist").removeClass("dno");
    } else {
        $(".image-exist").attr(
            ("src" " /media/"+data_Specie[$(this).val()].immagine).removeClass("dno");
        $(".image-not-exist").addClass("dno");
    }
    $("#nome_specie").html(data_Specie[$(this).val()].specie_padre)
    $("#nome_sottospecie").html(data_Specie[$(this).val()].nome);
    $("#nome_comune_sottospecie").html(data_Specie[$(this).val()].nome_comune);
    $(".taxonomy").removeClass("dno");
}
```

CARRELLO

#	Identificativo campione	Specie	Analisi	Attestati	Prezzo	
1	FOI 08 MX41 50 di Paolo Rossi	Uccellus Uccelli	smap	Cartaceo	€ 12,00	<a href="#">Elimina</a>
					spedizione attestati	€ 1,50
					Importo TOTALE come da listino	€ 13,50
					SCONTO APPLICATO	- € 0,00
					TOTALE SERVIZI <small>(iva inclusa)</small>	€ 13,50

INSERIMENTO CAMPIONE #1

Identificativo oppure RNA + Anello

FOI 08 MX41 50

Specie

Uccellus - Uccelli

Se non conosci il nome scientifico, clicca [QUI](#)

☐ Specie non trovata?

Mutazione

Mutazione (opzionale)

Analisi

= Attestato cartaceo

= Attestato digitale

Analisi

Attestati

Cartaceo

Digitale

♂ Sessaggio Molecolare

☒ SMAP (€ 10,00)  
da piuma, sangue, swab

☐ SMAU (€ 11,00)  
da uovo

☐ SMAR (€ 15,00)  
Rapido

Analisi Virali

☐ APV (€ 15,00)

☐ BFDV (€ 15,00)

Altre Analisi

☐ Clamidia (€ 20,00)

Immagine non disponibile

SPECIE: Uccellus

SOTTOSPECIE: Uccelli

NOMI COMUNI:

Scontil

Stai acquistando tutte le analisi SMAP su specie convenzionate a soli € 0,00!

Analisi selezionate:

• smap (€ 10,00)

(€ 2,00)

=

Totale campione: € 12,00

Data Nascita del soggetto

Proprietario

Nome e Cognome del Proprietario del soggetto

Inserimento Campione

Figura 4.6.: nuovo ordine - schermata carrello

23

ORDINE N° #12 -
ATTESA MATERIALE

CAMPIONI

Identificativo campione	Specie	Analisi / Stato / Attestati
FOI 08 MX41 50	Uccellus - Uccelli	SMAP <span>In attesa dei campioni</span> <span>Cartaceo</span>

COMUNICAZIONI


06/10/15-16:27

L'ordine #12 e' in attesa di materiale  
Il processo di creazione dell'ordine é stato completato. Modalità di pagamento scelta: Credito FEM.

Invia messaggio

Figura 4.7.: pagina ordine

Una volta completata l'aggiunta di soggetti si passa al *riepilogo*, sezione nella quale viene riproposto l'elenco completo degli esemplari aggiunti da analizzare, e richiesta l'eventuale aggiunta di campioni standard in caso di specie classificate come junior.

Infine si visualizza la schermata per il pagamento nella quale si sceglie attraverso **checkbox** la modalità preferita e dinamicamente vengono descritte in un `<div>` le coordinate per i pagamenti e per la spedizione delle piume.

Durante la creazione di un ordine può esserci un passo in più, ovvero nel caso di selezione di una specie non presente nell'elenco fornito da FEM2-Ambiente l'ordine viene arrestato, in attesa che un addetto dell'azienda classifichi correttamente la 'nuova specie' indicata dal cliente. Questo avviene sia per una questione di ordine e controllo, sia per mantenere la totale correttezza al momento della generazione degli attestati ed evitare di generare documentazione con specie non esistente e scritta in modo scorretto a causa di una svista.

## ordine

Completato l'ordine il cliente viene indirizzato al pagina dedicata, in cui vengono riproposte le coordinate di pagamento se non é ancora stato eseguito l'elenco dei soggetti da analizzare, lo stato delle analisi, la possibilità di scaricare gli attestati digitali e una sezione dedicata alla comunicazione diretta con FEM2-Ambiente (figura 4.7).



## 4.2.6. Vista Admin



Il controllo del flusso ordini e la comunicazione con il cliente sono una facciata importantissima per un e-commerce, e per questo motivo é necessaria una efficace interfaccia utente per l'utilizzo della piattaforma in tutte le sue potenzialità.

Django mette a disposizione un pannello admin autogenerato basato sui modelli definiti (vedi 4.1.3), ma per una migliore usabilità abbiamo deciso di installare *Django Suit*, una estensione per un tema alternativo del pannello admin dei sistemi Django [21]. Django Suit ci ha permesso di personalizzare il lato admin andando incontro alle richieste dei tecnici di FEM2-Ambiente. A lato si può vedere la barra verticale di navigazione.

Gli utenti lato admin, cioè dipendenti di FEM2-Ambiente, si possono dividere in due tipologie: gli addetti al controllo e ai rapporti con la clientela, e i tecnici di laboratorio.

I primi controlleranno le utenze, i dati della clientela, i prezzi e la scontistica legata o non legata alle associazioni convenzionate, le statistiche e la messaggistica con il

cliente.

I secondi si occuperanno di effettuare le analisi, controllare il corretto flusso degli ordini e la generazione degli attestati.

Il dettaglio delle azioni lato admin sono descritte nell'appendice B.

La configurazione di Django Suit avviene come segue

```
SUIT_CONFIG = {
    'ADMIN_NAME': 'Pannello admin del Portale Avifauna',
    'MENU': (
        'sites',
        {'label': 'Laboratorio', 'icon': 'icon-tint',
         'models': (
             'ordini.analisi',
             'ordini.metodoestrazione',
             'ordini.metodoamplificazione',
             'ordini.metodovisualizzazione',
             'ordini.campione')}),
    ),
}
```

## 5. Conclusioni

L'obiettivo dello stage é stato quello di sviluppare una piattaforma web per la creazione di un e-commerce personalizzato, facendo attenzione ad ogni parte del processo costruttivo.

La scelta del framework web é ricaduta su *Django* grazie alla sua flessibilità e alle sue proprietà; é stato in grado supportare la realizzazione una struttura complessa, aiutando gli sviluppatori con alcuni tool inclusi per la costruzione del pannello admin, la traduzione per una piattaforma multilingua e la creazione di file pdf e file testuali per l'esportazione di dati.

Per lo sviluppo del sito durante il flusso di creazione e gestione dell'ordine dal lato cliente si sono sfruttate le più famose tecnologie per lo sviluppo web, basandosi sui linguaggi *JavaScript*, *CSS*, *HTML*, e un framework come *Bootstrap*.

HTML alla base di tutte le pagine visitate connesso ai fogli di stile CSS per il sostegno alla struttura e al lato estetico; JavaScript per le funzionalità aggiuntive e per la creazione di un interfaccia più usabile ed intuitiva per il cliente finale.

Bootstrap é stato un ottimo supporto per il livello più alto della costruzione delle pagine web grazie al suo tool completo di regole CSS e JavaScript.

Per il alto divulgativo e di spiegazione delle funzionalità del sito la scelta é stata orientata su un CMS per fornire ai tecnici di FEM2-Ambiente uno strumento facile da usare senza la necessità di conoscere tecniche in informatica, così si é deciso per l'installazione di WordPress.

Il risultato é stato un sito fluido e funzionale che fonde una parte espositiva dei servizi forniti con la piattaforma di vendita vera e propria, nascondendo al cliente la complessità del sistema, ma contemporaneamente fornendo ai dipendenti di FEM2-Ambiente uno strumento completo per la gestione di tutti i componenti, dal sito, al flusso degli ordini, alla gestione delle analisi fino alla generazione di attestati finali.

# A. Codice Sorgente

In questa sezione sono riportate e commentate alcune parti del codice sorgente citate nel corso della relazione.

## Templates Django

Come illustrato nel corso della discussione Django fornisce uno strumento utile per centralizzare e non duplicare codice HTML, i *templates*.

Si è costruito un file `base.html` (di seguito parte del codice semplificato) per includere in un file unico tutte le informazioni statiche e ripetute in tutti gli altri file HTML del progetto.

```
<html>
<head>
  <title>{% block title %}fem2ambiente {% endblock %}</title>
  <link rel="stylesheet" type="text/css" href="/static/css/bootstrap.css">
</head>
<body>
  <nav role="navigation">
    {% include "login.html" %}
  </nav>
  <main>
    {% block content %}{% endblock %}
  </main>

  {% block jsexec %}
    <script src="/static/js/bootstrap.js"></script>
  {% endblock %}
</body>
</html>
```

Si può notare:

- nell'<head> il blocco <title> contiene `{% block title %}`, in modo da poter includere nel blocco un titolo diverso per ogni file HTML che viene esteso

- `{% include "login.html" %}` é la chiamata per includere in quella sezione della barra di navigazione il file dedicato al login, nel quale in base allo stato del cliente (loggato o no) mostrerà il form di login oppure i dati personali
- `{% block content %}` indica dove andrà inserito il codice HTML del file che estende
- `{% block jsexec %}` indica la sezione dedicata all'inserimento degli script JavaScript

Di seguito un esempio di estensione del file `base.html`.

```
{% extends "base.html" %}
{% block title %}
    {{ block.super }} - Pagina Personale
{% endblock %}
{% block content %}
    <div>[...]</div>
{% endblock %}
{% block jsexec %}
    {{ block.super }}
    <script type="text/javascript" src="/static/js/fledjed.js"></script>
{% endblock %}
```

Si può notare:

- nella prima riga `{% extends "base.html" %}` necessario per prendere tutte le informazioni di intestazione eccetera dal file di base
- in `{% block title %}` (come in `{% block jsexec %}`) il codice `{{block.super}}` che indica come viene ereditato il contenuto del blocco titolo presente nel file di base ed esteso con il contenuto in questo file (analogamente al significato di `super` in Java). Il titolo risultante sarà `fem2ambiente - Pagina Personale`.

## Traduzione

Django mette a disposizione un sistema per la gestione multilingua della piattaforma, per farlo é necessario inserire piccole parti di codice in tutti i punti che dovranno essere tradotti, sono dette *'translation strings'*. Django poi provvederà automaticamente ad incapsulare le stringhe da tradurre e trasferirle in file apposta dove andrà inserita la traduzione; una volta fatto ciò il sistema sarà pronto per supportare il multilingua.

Nel file `settings.py`:

```

LANGUAGES = (
    ('it', gettext('Italian')),
    ('en', gettext('English')),
    ('de', gettext('German')),
    ('fr', gettext('French')),
    ('es', gettext('Spanish')),
)

```

In ogni template HTML é necessario inserire la riga di codice `{% load i18n %}` per permettere a Django di individuare le stringhe da tradurre, inoltre esse dovranno essere più precisamente indicate.

Una soluzione é il tag `trans` utilizzato ad esempio

```
{% trans "stringa da tradurre" %}
```

oppure (se la stringa da tradurre é utilizzata più volte

```
{% trans "stringa da tradurre" as esempio_variabile %}
```

e all'interno del testo si andrà ad inserire la variabile collegata `{{esempio_variabile}}`.

Un'altra soluzione é l'utilizzo del tag `{% blocktrans %}`, di seguito un esempio di utilizzo

```

{% blocktrans %}
    stringa da tradurre nella quale si puo inserire un {{esempio_variabile}}
{% endblocktrans %}

```

Procedimento analogo va applicato ai file JavaScript che prevedono un output testuale che richiede traduzione, utilizzando la stringa `gettext()`

```
document.write(gettext('stringa da tradurre'));
```

Il risultato finale sono file generati da Django con l'estensione `.po` in cui viene indicata riga, file e path di ogni stringa da tradurre, la stringa inserita nel file originale e lo spazio per inserire la traduzione

```

#: path/del/file.py:23
msgid "Benvenuti nel mio sito."
msgstr "Welcome to my site."

```

Per riesaminare e controllare le stringhe da tradurre a terminale:

```
django-admin makemessages -a
```

e per compilare

```
django-admin compilemessages
```

Per i file JavaScript, per ogni lingua

```
django-admin makemessages -d djangojs -l it
```

## B. Guida Admin

Il pannello admin Django del Portale Avifauna deve permettere di controllare ogni passo del sistema.

### Utenze

Un utente del sistema é un entità identificata da un univoco indirizzo email, con password e nominativo. Esso ha attributi booleani per indicare se é attivo, se ha privilegi di staff o da superutente; é inoltre possibile indicare i singoli privilegi in un apposito elenco.

Un cliente invece é un entità associata ad un utente, con tutti gli attributi personali come nome, cognome, codice fiscale/partita iva, indirizzo, contatti telefonici e attributi di sistema come lo schema prezzi associato, la lingua oreferita, la quantità di crediti FEM in possesso e l'eventuale collegamento ad una associazione.

Le associazioni sono caratterizzate da un nome univoco, uno schema prezzi associato e eventuali informazioni aggiuntive.

Per indicare la correlazione tra cliente ed associazione esiste l'entità **iscrizione ad associazione** con il nominativo del cliente, l'associazione collegata e il numero di tessera corrispondente.

É stato inoltre aggiunto l'attributo booleano **ufficiale** per permettere agli addetti di FEM2-Ambiente di indicare quando l'iscrizione del cliente all'associazione corrisponde al vero, poiché il registro degli iscritti di ogni associazione é aggiornato continuamente ma inviato a FEM2-Ambiente solo ad intervalli temporali.



## Impostazioni ordini

In questa sezione si trovano quelle opzioni impostabili una tantum che non subiscono frequenti variazioni o controlli.

Vengono raccolti gli schemi di prezzi creati, indicati dal nome e contenenti le tariffe di ogni analisi e attestato; hanno la possibilità di essere di due tipi: Convenzioni o Pacchetti.

Nel primo caso (tipico di associazioni o clienti professionisti che analizzano un range di specie ridotto) vengono impostati i costi delle analisi come convenzionati, l'ulteriore costo scontato in caso di superamento di una soglia minima dell'ordine e la soglia da superare; vengono anche elencate le specie sulle quali effettuare i prezzi favorevoli e quali invece mantengono il prezzo di listino.

Nel secondo caso invece vanno impostati anche i prezzi scontati di tutte le analisi da applicare in caso di combinazione tra più analisi richieste sullo stesso soggetto.

Vengono anche indicate le cifre dei pacchetti crediti FEM acquistabili, ovvero il prezzo di ciascuno e il credito che il cliente accumula.

Infine si possono impostare e modificare i **template messaggi**, cioè i messaggi pre-popolati che possono essere utilizzati nelle fasi di invio comunicazioni automatiche. Ogni template ha un nome identificativo univoco, descrizione e ordinamento opzionali e i corpi del testo divisi per ogni lingua.

## Ordini

Gli ordini sono descritti da campi non modificabili manualmente come lo stato, l'ammontare e il cliente associato; contengono note fiscali o interne, e mostrano la lista di campioni associati con la possibilità di indicare eventuali problematiche di ogni singolo campione. Viene fornita la possibilità di assegnare un **idLab** (numero sequenziale per il laboratorio) ad ogni campione

# Bibliografia

- [1] FEM2 - Ambiente Srl, *Chi siamo*  
<http://fem2ambiente.com/it/chi-siamo.html>
- [2] ZooPlantLab  
<http://www.zooplantlab.btbs.unimib.it/>
- [3] Joomla!, *About Joomla!*  
<https://www.joomla.org/about-joomla.html>
- [4] VirtueMart, *What is VirtueMart?*  
<http://virtuemart.net/features/what-is-virtuemart>
- [5] WordPress, *About WordPress?*  
<https://wordpress.org/about/>
- [6] WordPress, *Installing WordPress*  
[http://codex.wordpress.org/Installing\\_WordPress](http://codex.wordpress.org/Installing_WordPress)
- [7] YOOtheme, *Company*  
<http://yootheme.com/company>
- [8] YOOtheme, *Wrap Framework*  
<http://yootheme.com/themes/warp-framework>
- [9] YOOtheme, *Uikit*  
<http://getuikit.com/>
- [10] Frédéric Demarle, *Polylang*  
<https://polylang.wordpress.com/>
- [11] Joseph C Dolson, *My Calendar*  
<https://www.joedolson.com/my-calendar/>
- [12] Django team, *About the Django Software Foundation*  
<https://www.djangoproject.com/foundation/>  
<https://docs.djangoproject.com/en/1.8/>



- [13] Federazione Ornicoltori Italiani - Onlus, *Chi siamo*  
<http://www.foi.it/la-federazione-ornicoltori-italiani-onlus.html>
- [14] World Wide Web Consortium, *W3C HTML, The Web's Core Language*  
<http://www.w3.org/html/>
- [15] World Wide Web Consortium, *Cascading Style Sheets (CSS)*  
<http://www.w3.org/TR/CSS/>
- [16] Standard ECMA-262, *ECMAScript® 2015 Language Specification*  
<http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [17] Bootstrap, *Designed for everyone, everywhere*  
<http://getbootstrap.com/>
- [18] Bootstrap - twbs, *Bootstrap - GitHub project*  
<https://github.com/twbs/bootstrap>
- [19] Font Awesome, *Font Awesome - The iconic font and CSS toolkit*  
<https://fontawesome.github.io/Font-Awesome/>
- [20] FontAwesome, *The iconic font and CSS framework*  
<https://github.com/FontAwesome/Font-Awesome>
- [21] Django Suit, *Modern theme for Django admin interface*  
<http://djangosuit.com/>  
<https://django-suit.readthedocs.org/en/develop/>