

## 1 LA SPERIMENTAZIONE IN CLASSE DI UN CASO CONCRETO

### 1.1 Il caso

La presente Unità è incentrata sulla realizzazione di un progetto software, commissionato nel 2019 da un'associazione di volontariato a una classe quinta di specializzazione informatica. Durante le ore della materia Gestione progetto, integrate con quelle di laboratorio per le altre materie di indirizzo, gli studenti hanno progettato e creato un'**applicazione vera e propria**, consegnata al committente e presentata all'esame di Stato.

Un simile progetto è valido come percorso di alternanza scuola-lavoro (PCTO) e può essere ulteriormente valorizzato sfruttando la sua **valenza multidisciplinare**. Se, infatti, l'ambito applicativo fa riferimento, per esempio, al volontariato o alla sostenibilità ambientale, è possibile integrare il progetto nelle materie umanistiche. La gestione dei dati e le statistiche coinvolgono invece la matematica. Infine, se l'associazione in questione opera fuori dai confini nazionali, la documentazione, la presentazione finale e l'interfaccia potrebbero essere realizzate in lingua straniera. Appare quindi evidente come sia possibile trovare nel progetto l'applicazione pratica delle altre materie di studio.

### 1.2 Il ciclo di vita del progetto

Ogni progetto è caratterizzato da un **ciclo di vita** costituito da fasi che si susseguono fino al raggiungimento degli obiettivi prefissati, come approfondito nell'Unità 4. Ciascuna fase comprende, oltre ai processi operativi, attività gestionali finalizzate a pianificare e controllare lo stato del progetto durante il suo sviluppo.

Ricordiamo che le **fasi del progetto** sono avvio, pianificazione, programmazione, esecuzione vera e propria (o realizzazione), controllo e chiusura.

Il passaggio da una fase all'altra avviene sulla base di criteri determinati, ma i processi coinvolti appaiono collegati tra loro, per cui i risultati di uno rappresentano gli input del successivo. Inoltre, le tre fasi centrali possono ripetersi ciclicamente.

Nel presente caso è stato impiegato un metodo di tipo iterativo detto **RUP** (Rational Unified Process), adattato per quanto possibile al contesto di un gruppo di studenti. Attraverso questo approccio, le fasi di pianificazione e programmazione vengono considerate come un'unica fase, detta "di approfondimento", e il progetto viene gestito come una serie di piccoli progetti sviluppati in modo iterativo e incrementale. In ogni iterazione, che sancisce uno step nel ciclo di vita del progetto, è necessario:

- selezionare e analizzare i casi d'uso rilevanti;
- effettuare la progettazione (design) sulla base dell'architettura scelta;
- implementare il design nei suoi componenti;
- verificare che i componenti soddisfino i casi d'uso.

Occorre precisare che la necessità di adeguare il progetto alla realtà del contesto può comportare scostamenti dalla metodologia di riferimento.

Nelle Lezioni seguenti sono riportati punti salienti e porzioni significative dei documenti prodotti nelle diverse fasi del progetto.

## 2 AVVIO

L'avvio rappresenta il riconoscimento formale dell'impegno a iniziare un progetto. In questa fase viene designato un **project manager** (PM) e viene predisposto un processo di **pianificazione** che permetta di raggiungere gli obiettivi prefissati.

### 2.1 L'idea e il bisogno

Un progetto deve necessariamente nascere da un'idea, che deriva in genere da una **necessità** in cui ci si imbatte. Le idee possono provenire dagli studenti, dai docenti, dalla scuola stessa o, come in questo caso, da una richiesta esterna.

#### caso

Dall'incontro con un'associazione di volontari è nata l'idea di **mappare** un'area della Guinea-Bissau (villaggio di N'Tchangue e zone limitrofe) e **realizzare un'applicazione web** per la gestione del censimento della popolazione che vi abita.

Un progetto deve poi soddisfare un **bisogno**. La digitalizzazione di un processo per rendere più veloci determinate procedure è di per sé una buona motivazione per un progetto informatico, ma è bene identificare il bisogno primario per cui è emersa l'idea.

#### caso

##### Mappa del territorio e anagrafe web

La mappatura dei villaggi in cui vivono migliaia di persone è importante per comprendere la composizione del territorio e la distribuzione degli agglomerati urbani che ne definiscono la morfologia. Poter disporre di una mappa condivisa come **#OpenStreetMap** (OSM) consente di far conoscere un territorio inizialmente sconosciuto al resto del mondo. Le mappe OSM, nella loro natura decentralizzata e collaborativa, offrono interessanti opportunità: gli studenti, per esempio, possono supportare le comunità locali nell'inserimento dei dati geografici (come case, moranças, ovvero gruppi di case, pozzi, strade, sentieri e qualsiasi altro oggetto visibile), appropriandosi così del contesto geografico e socio-culturale del luogo. Grazie all'app di navigazione è inoltre possibile orientarsi nei territori mappati. Poter disporre infine di un archivio consultabile via web dei dati sulla popolazione, con l'elenco degli abitanti, l'organizzazione delle famiglie con l'indicazione delle rispettive case, poterne visualizzare la posizione sulla mappa ed effettuare statistiche, è molto utile sia per l'amministrazione locale sia per i volontari che operano nel villaggio. Tali informazioni possono risultare inoltre utili in caso di emergenze sanitarie (per esempio pandemie) o calamità naturali.



Guarda la **posizione del villaggio** su OpenStreetMap.

#### #techwords

##### OPENSTREETMAP (OSM)

È un progetto collaborativo finalizzato a creare un database mondiale di dati geografici, editabile liberamente.

### 2.2 Il contratto

Una volta definiti l'idea e il bisogno che muovono il progetto, è importante stabilire un **patto condiviso**, in questo caso tra il committente, i docenti e gli studenti. In tale contratto si definiscono gli obiettivi, i tempi, i modi di svolgimento del progetto e le responsabilità di ogni componente del gruppo di lavoro (stakeholder).

## CASO

Il progetto OSMGB è stato proposto dall'associazione Onlus Abala Lite ai docenti della materia GPOL e alla classe quinta dell'indirizzo informatico di un Istituto di istruzione superiore nel settembre 2019 (A.S. 2019/20).

Gli allievi e i docenti si sono accordati per sfruttare l'opportunità data dalla commessa e applicare a un caso concreto i concetti acquisiti nelle lezioni teoriche, che riguardano la gestione di un progetto informatico in tutte le sue fasi. L'obiettivo non è dunque limitato alla consegna dell'applicazione per il sistema anagrafico e catastale, ma comprende anche l'**esperienza dell'apprendimento** attraverso una metodologia project-based: lavorare su un progetto reale, infatti, mette lo studente in relazione con clienti e consulenti e lo pone di fronte alle criticità delle varie fasi del progetto, costringendolo a mettere in pratica i contenuti delle lezioni frontali dei docenti. Ogni studente ha inoltre modo di sperimentare vari ruoli nel corso del progetto.

La trasversalità del progetto informatico costituisce infine un'opportunità per approfondire e sperimentare mano a mano concetti, contenuti e competenze trasversali alle varie materie di studio, da quelle tecniche a quelle umanistiche e linguistiche.

## 2.3 Il kick-off

Il kick-off rappresenta il momento di **avvio vero e proprio** del progetto. In genere si esplicita con una **riunione formale** per ufficializzarlo. Da questo momento l'idea non subisce più trasformazioni, gli obiettivi e la direzione da perseguire sono stati definiti e l'impresa prende vita.

## 2.4 Il project chart

Il project chart è un documento redatto nella fase di avvio che ha lo scopo di **ufficializzare il momento di inizio** del progetto e ne chiarisce alcuni aspetti gestionali (vedi Unità 4). È l'atto formale con cui le parti interessate vengono informate non solo dell'esistenza del progetto, ma anche di tutte le relative **caratteristiche essenziali**, compresa una prima stima delle responsabilità, degli incarichi e dei ruoli coinvolti.

Di seguito è esemplificato il project chart relativo al caso in esame.

## CASO

| 1. DATI GENERALI DI PROGETTO |  |
|------------------------------|--|
| Nome                         | OSMGB-Anagrafe Web   |
| Descrizione                  | Applicazione web per gestire i dati anagrafici della popolazione del villaggio di N'Tchangue in Guinea-Bissau e mappare su OpenStreetMap strade, case, morançās, pozzi, sentieri e altro |
| Project manager              | Paolo Rossi  |
| Team di lavoro               | Gruppo classe, docenti, consulenti tecnici   |
| Cliente                      | Abala Lite Onlus (associazione di volontariato)  |

**2. BENEFICI PREVISTI**

Per gli studenti:

- simulare un sistema lavorativo complesso;
- utilizzare conoscenze informatiche apprese negli anni scolastici;
- sperimentare il lavoro di gruppo;
- sperimentare conoscenze e competenze nella mappatura e progettazione web;
- sperimentare un percorso di PCTO.

Per il committente:

- il progetto è utile per gli abitanti locali e i volontari per disporre di un'anagrafe web e poter visualizzare sulla mappa OSM strade, case e altri elementi.

**3. PRINCIPALI OBIETTIVI**

- Migrare su un database relazionale i dati del censimento attualmente presenti in Excel.
- Creare un sistema web di gestione anagrafica integrato con OpenStreetMap, funzionale e scalabile.
- Mappare una zona specifica della Guinea-Bissau.

**4. METODOLOGIA E TEMPISTICHE**

Metodologia RUP di tipo incrementale e metodologia agile.

**Avvio:** ottobre - novembre

**Approfondimento:** ottobre - febbraio

**Realizzazione:** novembre - maggio

**Chiusura:** aprile - giugno

**5. DELIVERABLE DI PROGETTO**

|   |  |
|---|--|
| <b>Documenti del PM</b>                 | Piano di progetto (GANTT, WBS, RAM, misure, ecc.)  |
| <b>Analisi dati</b>                     | Analisi tabelle Excel e verifica consistenza dati, progettazione database (concettuale e logica)     |
| <b>Migrazione dati anagrafici su DB</b> | Pulizia e normalizzazione dei dati presenti su tabelle Excel e migrazione su database relazionale    |
| <b>Analisi dei requisiti</b>            | Documento con i principali requisiti utente  |
| <b>Analisi dei casi d'uso</b>           | Documento con i principali scenari e casi d'uso  |
| <b>Database</b>                         | Documento di progettazione del DB relazionale e creazione DB   |
| <b>Inserimento dati</b>                 | Creazione algoritmi per la migrazione automatica dei dati  |
| <b>Applicazione web</b>                 | Documentazione tecnica e rilasci successivi dell'applicazione web di anagrafica, con test e collaudo |
| <b>Mappatura</b>                        | Mappatura su OSM di case, pozzi, sentieri, strade, ecc. del villaggio di N'Tchangue (Guinea-Bissau)  |

**2.5 L'analisi di fattibilità**

È fondamentale verificare sempre la **fattibilità** di un progetto in base a costi, benefici, tempistiche, disponibilità di tecnologie, competenze, ecc. Affinché un progetto sia fattibile, è bene tenere a mente le seguenti regole fondamentali:

- **evitare complicazioni** e proseguire solo quando si ha chiaro che cosa fare;
- **suddividere il progetto** in versioni autonome e funzionanti, risolvendo di volta in volta i problemi e aggiungendo gradualmente funzionalità;
- impostare l'architettura di base e iniziare con dei **prototipi funzionanti**, affinando sempre più le successive versioni (release).

Nel caso di un progetto scolastico, può essere sufficiente chiedersi se i tempi previsti e le competenze necessarie sono compatibili con le richieste del committente.



## 2.6 I rischi

Come abbiamo visto nell'Unità 3, Lezione 6, è buona norma individuare e documentare fin da subito i **rischi** per cui un progetto potrebbe non avere successo, insieme alle **strategie** da mettere in atto per ridurli.

### CASO

#### #techwords

##### TRAINING ON JOB

Si tratta di un tipo di formazione erogata sul posto di lavoro in cui i dipendenti vengono fatti familiarizzare con l'ambiente e con ciò con cui dovranno interagire (macchinari, materiali, ecc.).

| Fattori di rischio  | Strategie   |
|---|---|
| Difficoltà nelle fasi iniziali su come partire                    | Riunioni con i docenti, utilizzo di schemi, brainstorming, richiesta di consigli a esperti, ecc.  |
| Mancata competenza tecnica degli studenti all'inizio del progetto | Corsi di rinforzo con i docenti o tra pari, autoformazione, <b>#training on job</b> , suddivisione dei ruoli e condivisione dei problemi, richiesta di supporto a esperti o consulenti. |
| Difficoltà nella comunicazione con il committente in Africa       | Utilizzo di riunioni online, email, incontri con conoscitori della società africana, ecc.   |
| Mancato rispetto dei tempi  | Riunioni periodiche sullo stato dei lavori (SAL), verbali di riunione, monitoraggio continuo (Unità 4, Lezione 10)  |
| Mancata conoscenza del dominio applicativo                        | Riunioni periodiche con il committente, con i volontari che operano localmente o con esperti del settore  |

## 2.7 Nomi e identità

La scelta del **nome** dà al gruppo di lavoro un'**identità**; a volte può essere stimolante anche trovare un **motto**. In un progetto scolastico questi accorgimenti permettono di definirsi come una vera azienda e di mettere in moto il processo di team building, osservato nell'Unità 4, Lezione 8. È utile che tutti i partecipanti collaborino nella ricerca del nome, per esempio in un brainstorming seguito da votazione, con validazione del PM o del docente.

### CASO

**Il nome:** OSMGB-Anagrafe Web.

**Il motto:** "Abala lite", ossia "come stai" in lingua balanta, parlata in Guinea-Bissau.

## 2.8 L'elevator pitch

Affinché il progetto ottenga l'approvazione del committente e della comunità interessata, come abbiamo visto nell'Unità 4, Lezione 11, è importante un'efficace comunicazione di progetto. In particolare, in questa fase l'idea può essere esposta attraverso l'**elevator pitch**, ossia un breve video o una presentazione in grado di **catturare l'attenzione** e **convincere** lo spettatore che il progetto è valido e vale la pena investire tempo e risorse.

### CASO

Gli studenti possono creare un breve video in cui loro stessi si chiariscono le idee su che cosa vogliono realizzare, in modo da poterlo raccontare in pochi secondi.

#### FISSA LE CONOSCENZE

1. Che cos'è il kick-off di un progetto?
2. Che cosa si intende per analisi di fattibilità?

## 3 APPROFONDIMENTO

### 3.1 L'analisi del contesto

Prima di iniziare a sviluppare un progetto è importante approfondire il contesto in cui sarà inserito: per evitare conseguenze imprevedibili e il rischio di insuccesso è infatti necessario conoscere **fattori contestuali** come le caratteristiche socioculturali e linguistiche dell'utenza finale del sistema, nonché i **bisogni** e i **vincoli** dei committenti. Un mezzo utile per raccogliere questi dati è rappresentato dall'**#intervista**.

#### #techwords

##### INTERVISTA

È un metodo di raccolta dei dati basato su un dialogo in cui l'intervistatore pone delle domande su un particolare tema a una o più persone selezionate.

#### CASO



La **Guinea-Bissau** è una tra le più piccole nazioni dell'Africa occidentale (36 125 km<sup>2</sup>), con capitale Bissau (600 000 abitanti). Si affaccia sull'Oceano Atlantico tra il Senegal a nord e la Repubblica della Guinea a sud-est. È il **quarto paese più povero** del mondo: il 69% della popolazione si trova nella fascia di povertà, il che significa che 1 075 082 persone vivono con meno di due dollari al giorno e il 33% vive in condizioni di estrema povertà con meno di un dollaro al giorno. Inoltre, si trova al 175° posto su 189 paesi secondo

il Rapporto globale sullo sviluppo umano del 2020. La mortalità infantile è molto alta: delle nuove nascite il 10% muore prima di aver raggiunto il primo anno, mentre un altro 10% non arriva a 5 anni. Chi sopravvive ha una speranza di vita di circa 50 anni secondo CIA World Factbook (dati aggiornati al gennaio 2015).

La guerra civile del 1998-1999 ha danneggiato ulteriormente un'economia già **fragile**, basata perlopiù su agricoltura e pesca. A causa della mancanza di infrastrutture e mezzi finanziari, infatti, la popolazione non è in grado di sfruttare le risorse minerarie presenti sul territorio (petrolio, bauxite e fosfati).



L'associazione **Abala Lite Onlus** ha avuto i primi contatti con il villaggio di N'Tchangue nel 2013. Il villaggio di **N'Tchangue** appartiene a Nhacra, provincia della regione Oio. È situato a circa 40 km a nord-est della capitale Bissau e a circa 20 km da Mansôa, cittadina di circa 45 000 abitanti che ospita quotidianamente un mercato coperto e che si trova sulla strada nazionale che da Bissau porta in Guinea e in Senegal.

Il villaggio si trova nella savana, a circa 4 km dalla strada nazionale che collega le due città.

In base al censimento del 2009, il villaggio di N'Tchangue è suddiviso in due distretti: N'TchangueBedeta e N'TchangueSubal.

Prima dell'avvio del progetto **non vi era alcun archivio informatizzato** per censire la popolazione e le proprietà del villaggio. Le informazioni erano tramandate oralmente, con conseguente rischio di conflittualità e incomprensioni tra gli abitanti. Un primo censimento è stato effettuato nel 2016 dai maestri di scuola, tramite interviste agli allievi e ai genitori.

Il frutto di questo lavoro è stato raccolto in un file Excel.

## 3.2 I verbali

I **verbali** sono documenti molto utili per tener traccia della storia delle attività intraprese e da svolgere.

Nella fase iniziale del progetto i verbali servono a documentare le **decisioni più importanti**, soprattutto in merito all'**assegnazione delle attività** e delle responsabilità. Nel corso del progetto tutti questi documenti devono essere mantenuti **aggiornati**, in particolare a seguito delle riunioni più importanti in cui sono prese decisioni significative.

Il verbale deve essere semplice e lineare: normalmente inizia con la data di svolgimento della riunione, la lista di partecipanti e il nome del redattore del documento. Nel caso del progetto di classe, è consigliabile che il redattore cambi a ogni riunione, così che ciascuno studente possa provare in prima persona l'attività e capirne l'importanza. Inoltre, è utile come pratica alla presa di responsabilità in forma scritta.

Ogni verbale deve essere condiviso con tutto il gruppo di lavoro e quindi è utile, per esempio, inviarlo via email poco dopo la riunione stessa con richiesta di approvazione.

### CASO

Progetto OSMGB      Verbale n. 01

Luogo: \_\_\_\_\_, data, ora di svolgimento: \_\_\_\_\_

Partecipanti: \_\_\_\_\_ (elenco dei partecipanti)

Obiettivo dell'incontro: \_\_\_\_\_ (breve sintesi dei motivi della riunione)

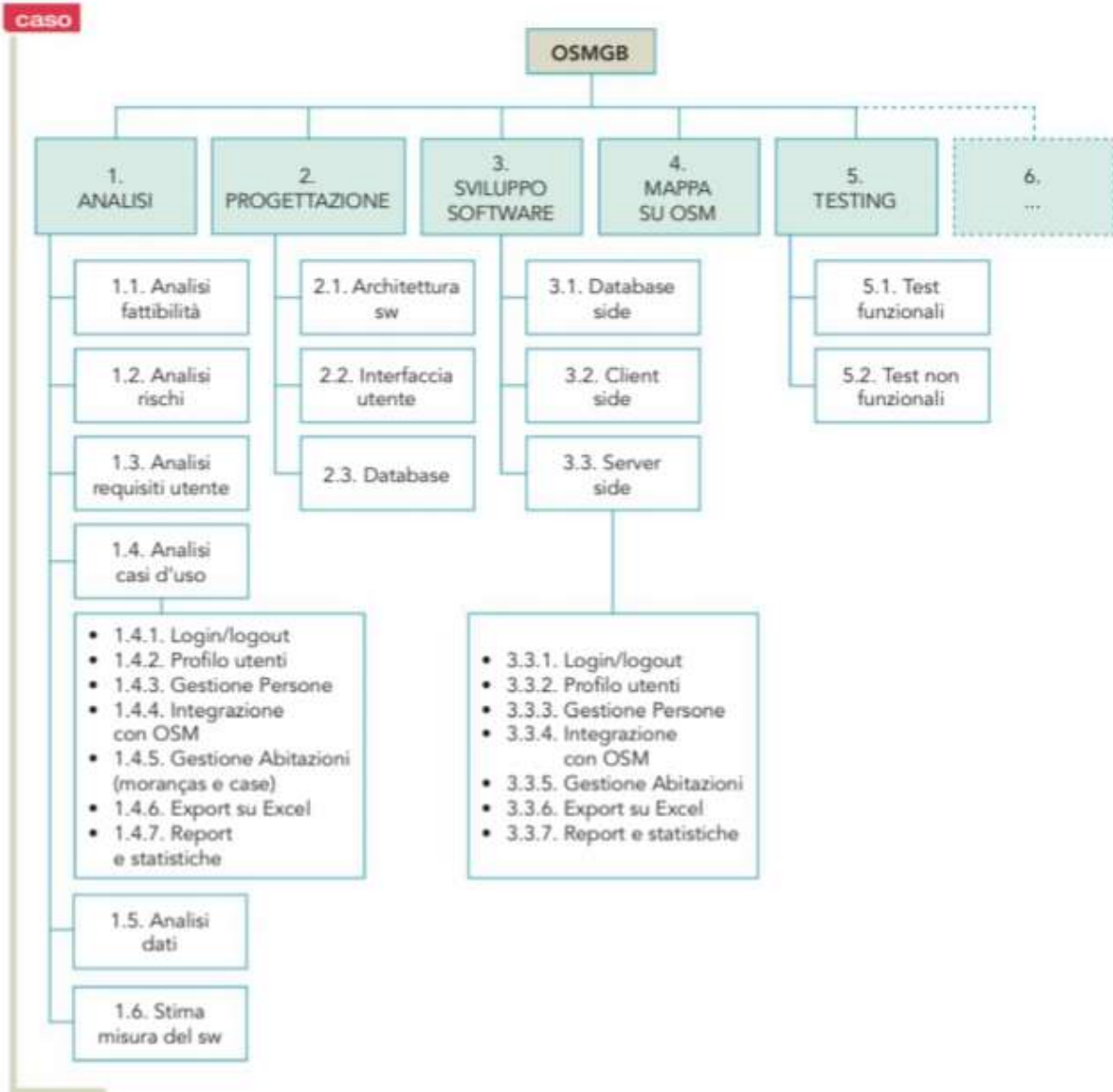
Principali punti emersi: \_\_\_\_\_

Decisioni prese e prossimi impegni: \_\_\_\_\_

## 3.3 La WBS

Come visto nell'Unità 4, Lezione 2, la WBS consiste nella scomposizione gerarchica dell'intero progetto in fasi di lavoro: il progetto, che costituisce la radice della struttura ad albero, viene scomposto in componenti e sottocomponenti sempre più piccoli e semplici da gestire, i cosiddetti work package, finché le loro dimensioni non sono tali da permettere una stima sufficientemente accurata sui relativi tempi e costi e l'assegnazione di un responsabile.

Di seguito è riportata parte della WBS di progetto.



### 3.4 Il diagramma di Gantt

Per legare la WBS alle tempistiche e alle risorse, è utile ricorrere al diagramma di Gantt. Come descritto nell'Unità 4, Lezione 5, si tratta della rappresentazione su un piano cartesiano in cui sull'asse verticale vengono riportate le attività della WBS, mentre lungo l'asse orizzontale si ha la variabile temporale (tipicamente espressa in giorni di lavoro). Ciascuna attività viene quindi rappresentata mediante una barra orizzontale, la cui posizione è determinata dalla data di inizio prevista e la cui lunghezza rappresenta la durata stimata in sede di previsione.



Questo tipo di diagramma permette al project manager di verificare in ogni momento lo **stato di avanzamento** del progetto rispetto alle previsioni iniziali, i costi risultanti, le tempistiche delle varie attività e l'allocazione delle risorse del gruppo di lavoro. Di seguito si trova una porzione del diagramma di Gantt del progetto, utilizzato per la suddivisione delle attività.

**caso** **DURATA** Usando ProjectLibre si può inserire per ciascun WP una durata. In automatico il software determina la durata complessiva della macroattività.

|    |  | Nome                                   | Durata    |
|----|--|--|-----------|
| 1  |  | Inizio progetto                        | 0 giorni  |
| 2  |  | 0.PIANIFICAZIONE                       | 5 giorni  |
| 3  |  | 1.ANALISI                              | 76 giorni |
| 4  |  | 1.1.Analisi fattibilità                | 10 giorni |
| 5  |  | 1.2.Analisi rischi                     | 3 giorni  |
| 6  |  | 1.3.Analisi requisiti utente           | 10 giorni |
| 7  |  | 1.4.Analisi Casi d'Uso                 | 46 giorni |
| 8  |  | 1.4.1.Analisi Login/Logout             | 10 giorni |
| 9  |  | 1.4.2.Analisi Gestione profilo Utenti  | 10 giorni |
| 10 |  | 1.4.3.Analisi Gestione Persone         | 10 giorni |
| 11 |  | 1.4.4.Analisi Integrazione con OSM     | 15 giorni |
| 12 |  | 1.4.5.Analisi Gestione Abitazioni      | 40 giorni |
| 13 |  | 1.4.6.Analisi Gestione Export su excel | 20 giorni |
| 14 |  | 1.4.7.Analisi Report e Statistiche     | 15 giorni |
| 15 |  | 1.5.Analisi Dati                       | 15 giorni |
| 16 |  | 1.6.Stima misura del software          | 10 giorni |

**NUMERAZIONE** ProjectLibre numera in automatico e in modo progressivo ciascuno dei WP inseriti nel progetto.

### 3.5 La gestione della documentazione di progetto

Esistono numerosi strumenti che consentono di gestire e condividere la documentazione del progetto tra i partecipanti. Uno dei metodi più semplici, però, è quello di creare uno **spazio di archiviazione condiviso** a cui tutti possono accedere, idealmente in cloud. Nel caso in questione, potrebbero essere il server della scuola, Google Drive oppure altri servizi gratuiti.

L'organizzazione più semplice è quella ad albero, con una radice (root) in cui di norma si crea un **#documento master** di progetto contenente le informazioni più importanti, il project chart e le sottocartelle organizzate in base alla scelta del project manager.

#### #techwords

##### DOCUMENTO MASTER

È un documento cui sono associati dei documenti secondari.

#### esempio

Possibili cartelle, a loro volta suddivisibili in sottocartelle, sono:

- verbali, contenente i verbali di progetto;
- requisiti, che ospita i documenti con i requisiti di progetto;
- pianificazione, contenente i documenti di pianificazione;
- documenti\_tecnici, che raccoglie la documentazione tecnica.

### 3.6 Il gruppo di lavoro e le responsabilità

In fase di approfondimento del progetto è necessario anche definire il gruppo di lavoro, i ruoli e le singole responsabilità.

#### esempio

Tipicamente in un progetto software si trovano diversi ruoli:

- |                               |                               |
|-------------------------------|-------------------------------|
| ■ project manager;            | ■ esperto back-end;           |
| ■ esperto database;           | ■ esperto di test e collaudo; |
| ■ esperto di migrazione dati; | ■ analista;                   |
| ■ esperto statistiche;        | ■ sistemista;                 |
| ■ esperto front-end;          | ■ esperto di sicurezza.       |

Non è detto che ciascun componente del gruppo ricopra un solo ruolo. L'esperto di database potrebbe anche rivestire il ruolo di esperto di statistiche, così come potrebbero esserci due o più analisti.

Nel contesto della classe, è importante che l'**organigramma** del gruppo di lavoro che si viene a creare, che simula quello di un'azienda, sia **condiviso** da tutti. Per ogni componente del gruppo deve essere predisposto fin da subito anche un eventuale ruolo alternativo o secondario.

È consigliabile non modificare nel tempo i ruoli, a meno che non sussistano situazioni particolari o la volontà di sperimentare ruoli diversi.

In un gruppo di studenti possono interagire personalità differenti. Di norma le varie figure (il leader, il tecnico, il grafico, il commerciale, ecc., oltre ai ruoli sopraelencati relativi allo sviluppo di software) nascono e si configurano quasi spontaneamente nel tempo. Il concetto di **responsabilità**, ovvero essere abili a rispondere del proprio ruolo nel progetto, è fondamentale: ogni singolo partecipante deve sentirsi responsabile sia individualmente che collettivamente del lavoro che ha deciso di intraprendere insieme ai suoi compagni.

Per definire già nel project chart un primo organigramma delle figure professionali coinvolte, può essere utile utilizzare la RAM (matrice compiti/responsabilità), descritta in dettaglio nell'Unità 4. La matrice, detta anche RACI (Responsible, Accountable, Consulted, Informed), è una rappresentazione dei ruoli in relazione al progetto.

Tutte le figure professionali, individuate dagli "attori", che operano all'interno del progetto vengono infatti associate alle varie attività in base ai seguenti criteri.

- 'R': l'attore è attivamente coinvolto nell'attività mediante una **responsabilità di tipo operativo**, che può essere o meno condivisa con altri membri del team.
- 'A': l'attore è il **responsabile ultimo** dell'attività, si occupa di approvare il lavoro dei ruoli 'R' e ha l'ultima parola sulle decisioni che riguardano l'attività. Di norma il ruolo non è condiviso.
- 'C': l'attore viene **consultato** in merito a come si svolge l'attività. Il ruolo può essere assegnato contemporaneamente a più membri del team.
- 'I': l'attore (o gli attori) deve essere **informato** sullo stato dell'attività.

Nella pagina successiva è riportata parte di una RAM relativa a sette attori.

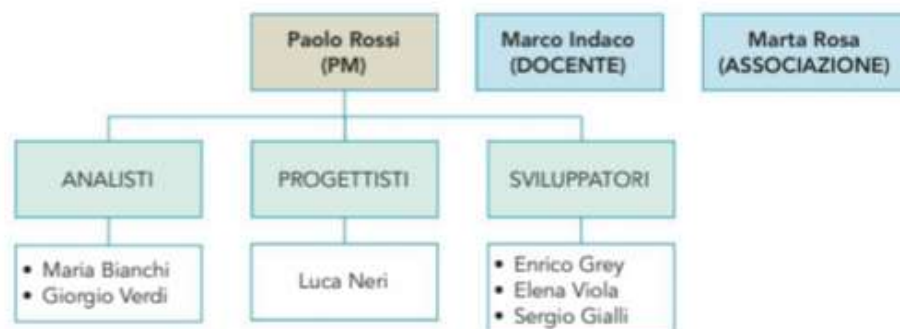
CASO

| Attività                 | Attori (ruolo principale) |                          |                          |                         |                           |                         |                         |
|--------------------------|---------------------------|--------------------------|--------------------------|-------------------------|---------------------------|-------------------------|-------------------------|
|                          | Paolo Rossi (PM)          | Maria Bianchi (Analista) | Giorgio Verdi (Analista) | Luca Neri (Progettista) | Sergio Gialli (Developer) | Elena Viola (Developer) | Enrico Grey (Developer) |
| Analisi fattibilità      | A                         | R                        | C                        | C                       | C                         | C                       | C                       |
| Analisi rischi           | C                         | C                        | A                        | R                       | I                         | I                       | I                       |
| Analisi requisiti utente | C                         | C                        | A                        | C                       | C                         | R                       | R                       |
| Login/logout             | A                         | R                        | I                        | I                       | I                         | I                       | I                       |
| Profilo utenti           | C                         | A                        | I                        | I                       | I                         | I                       | R                       |
| Gestione persone         | R                         | A                        | C                        | C                       | C                         | I                       | I                       |
| Integrazione con OSM     | C                         | C                        | I                        | I                       | A                         | R                       | I                       |
| Gestione abitazioni      | R                         | I                        | I                        | I                       | I                         | I                       | A                       |
| Export su Excel          | C                         | C                        | R                        | A                       | C                         | C                       | C                       |
| Report e statistiche     | I                         | I                        | I                        | I                       | R                         | A                       | I                       |

Tramite l'uso della RAM è possibile definire in modo chiaro e preciso le relazioni tra gli individui o le unità e quindi i compiti e le responsabilità relativi. Da ciò si può ricavare anche una stima **percentuale del coinvolgimento** dei singoli attori per ogni attività. Per esempio, un ruolo R solitamente sarà impegnato al 100% (o quasi) sull'attività, i ruoli A e C con una percentuale minore e I con percentuale nulla.

In fase di avvio del progetto avviene la definizione del gruppo di lavoro, dei ruoli e delle singole responsabilità (FIGURA 1). Oltre al project manager, il primo responsabile del progetto, ogni attore ha delle **responsabilità individuali** (ovvero riguardo al ruolo particolare che svolge) ma anche **collettive**, che possono influire sul progetto intero o su altri individui. Per esempio, se un analista produce un documento di analisi con degli errori, ciò si ripercuoterà a cascata sulle fasi successive (come lo sviluppo del software o la progettazione del database) e su altri attori.

FIGURA 1 Definizione del gruppo di lavoro, ruoli e responsabilità



FISSA LE CONOSCENZE

1. A che cosa serve redigere i verbali di riunione?
2. Che cos'è il diagramma di Gantt?
3. Perché è importante documentare un progetto?

## 4 REALIZZAZIONE

### 4.1 Il monitoraggio del progetto

#### ■ Le milestone

Un progetto è caratterizzato da un proprio ciclo di vita che è necessario monitorare costantemente, definendo delle **tappe intermedie** di controllo dell'avanzamento delle attività.

Le tappe più importanti, come abbiamo visto, sono dette **milestone**, ovvero pietre miliari. Per esempio, nel caso di un'applicazione, possono essere la definizione dei requisiti utente, la validazione da parte del committente, la progettazione del database, le fasi di rilascio intermedie, i test, ecc.

#### ■ Lo Stato Avanzamento Lavori

Numerose tappe intermedie prevedono l'uso dei cosiddetti **Stati Avanzamento Lavori** (SAL): si tratta di brevi **riunioni** che coinvolgono il gruppo di lavoro, a cadenza settimanale o mensile, in cui ci si confronta sull'andamento generale del progetto. In tali occasioni viene redatto un verbale da archiviare.

### 4.2 L'analisi dei requisiti utente

Nella prima fase di realizzazione è importante prendere in considerazione i principali **requisiti utente**, ovvero le richieste che il sistema dovrà soddisfare per l'utente finale. Per identificarle è necessario svolgere numerosi incontri con il committente e gli utenti finali del sistema.

#### CASO



Come detto in precedenza, la mappatura del territorio sarà effettuata su OpenStreetMap, una piattaforma open source collaborativa che permette di inserire su un vero e proprio database reticolare oggetti di varie tipologie, come case, sentieri, pozzi e strade.

Anagrafe Web è l'applicazione web appoggiata a un database relazionale che permette di censire la popolazione del villaggio, identificando i singoli abitanti, le case e le morançās, ossia i raggruppamenti di case, in cui vivono i

nuclei familiari. I dati iniziali possono essere inseriti manualmente o possono essere recuperati dal file Excel prodotto dal primo censimento.

Il sistema deve inoltre permettere un'interazione tra il sistema web gestionale e la mappa OSM, in modo da identificare, per esempio, a partire dai dati di un determinato abitante, la posizione della casa in cui abita e, viceversa, centrando sulla mappa una casa, risalire agli abitanti che la occupano.



Leggi  
l'**approfondimento**  
online su  
OpenStreetMap



## #techwords

## LINGUAGGIO UML

È un linguaggio di modellazione e di specifica basato sul paradigma orientato agli oggetti.

I principali requisiti utente possono essere dettagliati ulteriormente attraverso gli **scenari di utilizzo**. Questi sono rappresentabili anche graficamente con i diagrammi dei casi d'uso del **#linguaggio UML** (Unified Modeling Language). I **casi d'uso** corrispondono alle principali funzionalità di cui potrà usufruire l'utente che utilizzerà il sistema informatico e possono essere classificati in due tipologie:

- **funzionali**, strettamente legati alle funzionalità viste dall'utente che utilizza il sistema;
- **non funzionali**, che non sono visibili chiaramente dall'utente finale, ma devono essere identificati e analizzati per garantire la sicurezza, la qualità, la correttezza e l'efficienza del sistema.

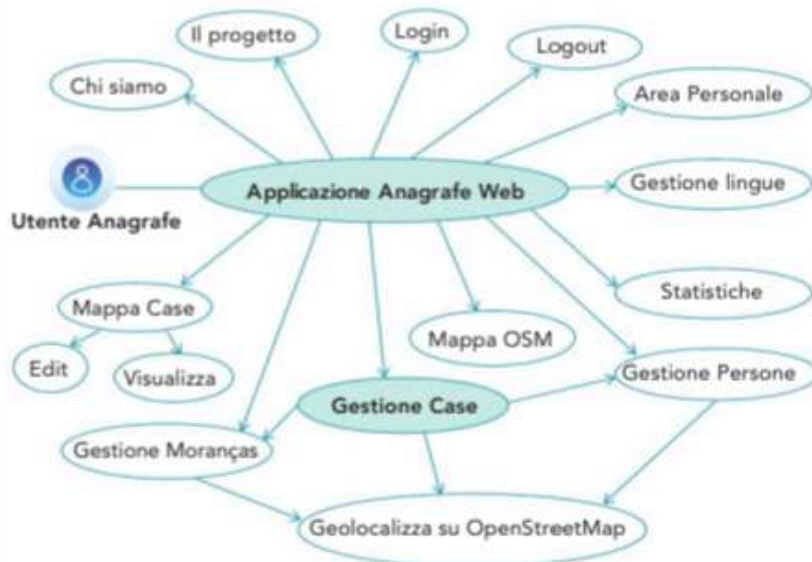
## caso

Il villaggio N'Tchangue è formato da raggruppamenti di case, cioè le moranças, in cui vivono delle **persone** organizzate in **famiglie**.

Una famiglia è composta da un insieme di persone, ognuna delle quali ha un ruolo o **grado di parentela** rispetto al **capo famiglia**. Lo stato civile di una persona può variare nel tempo, in base a determinati eventi che possono accadere (nascita, morte, matrimonio, separazione, ecc.). Anche la composizione di una famiglia può variare nel tempo, in base ad altri eventi (nuovo insediamento, cambio di casa, fusione con altra famiglia, ecc.). Un capo famiglia può, per esempio, avere più mogli nel corso del tempo o può integrare nella sua famiglia altre mogli provenienti da altre famiglie. In tale contesto è necessario quindi memorizzare la storia degli stati civili delle persone e della composizione delle famiglie.

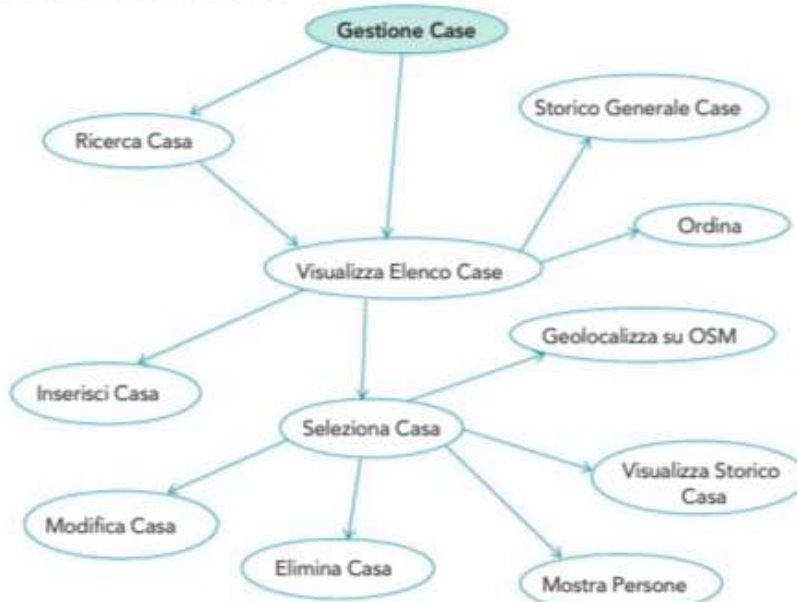
## Caso d'uso generale dell'applicazione Anagrafe Web

Nel diagramma che segue sono evidenziati i principali casi d'uso dell'applicazione Anagrafe Web. Ogni caso d'uso corrisponde di fatto a una voce di menù o funzionalità vista dall'utente che utilizza l'applicazione.



**Gestione case**

Nel seguente diagramma è invece possibile osservare in dettaglio i casi d'uso relativi alla voce Gestione Case.



- **Ricerca Casa:** viene effettuata la ricerca di una casa, mediante filtri sul nome o sulla zona.
- **Seleziona Casa:** a fronte della ricerca, permette la selezione di una casa specifica.
- **Ordina:** modifica le opzioni di ordinamento delle case in elenco.
- **Visualizza Elenco Case:** sulla base della ricerca effettuata, visualizza l'elenco delle case, con i rispettivi dati.
- **Inserisci Casa:** permette di inserire una nuova casa.
- **Elimina Casa:** permette di eliminare la casa selezionata.
- **Modifica Casa:** modifica i dati relativi alla casa selezionata (con possibilità di caricare una foto).
- **Mostra Persone:** visualizza l'elenco delle persone che abitano la casa selezionata.
- **Visualizza Storico Casa:** visualizza la storia delle variazioni della casa selezionata.
- **Geolocalizza su OSM:** viene aperta una nuova pagina web su OpenStreetMap, nella quale viene centrata e visualizzata la casa selezionata.
- **Storico Generale Case:** visualizza lo storico, cioè la storia delle variazioni delle case.

### 4.3 L'interfaccia utente

Come per i casi d'uso, è altresì possibile definire le **#videate** dell'applicazione in maniera più o meno dettagliata. In una fase iniziale del progetto, quando è ancora necessario definire i campi da visualizzare, le schermate saranno costituite semplicemente da testo, senza particolari accorgimenti grafici. Man mano che si procede con le fasi di realizzazione, le videate potranno avvicinarsi sempre più all'interfaccia utente definitiva.

#### #techwords

##### VIDEATA

L'insieme dei dati e delle operazioni che possono essere visualizzati in una sola volta sullo schermo.

## CASO

## Login

Per effettuare il login al sito è necessario recarsi nel menù iniziale e selezionare la voce **Accedi** (icona con la porticina), come rappresentato in figura.



Dopodiché viene mostrata all'utente una pagina dove poter inserire i dati per effettuare l'accesso.



I dati richiesti per il login sono username e password. Per effettuare l'accesso, questi devono essere presenti sul database, secondo le norme previste per la sicurezza e la privacy. In caso di esito positivo nel controllo della corrispondenza, l'utente potrà accedere all'interno del sito; altrimenti, se l'utente sbaglia per 3 volte consecutive, dovrà aspettare 20 secondi prima di riprovare a effettuare l'accesso. La procedura è gestita tramite una tabella di login in cui vengono memorizzati gli ultimi tentativi.

## Gestione case

Scegliendo dal menù principale la voce **Case**, verrà mostrata all'utente una pagina contenente l'elenco delle case con le rispettive informazioni.



| Foto | Nome    | Id | Zona | Morança | Id morança | Capo famiglia | Id capo famiglia | N. abitanti | sulla mappa | data inizio val. | Modifica | Elimina | Personale | Storico |
|------|---------|----|------|---------|------------|---------------|------------------|-------------|-------------|------------------|----------|---------|-----------|---------|
|      | CASA 1  | 1  | NORD | ANDINE  | 1          | Antonio Rossi | 1                | 7           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 2  | 2  | NORD | ANDINE  | 1          | Piero Rossi   | 2                | 5           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 3  | 3  | NORD | ANDINE  | 1          | Roberto Rossi | 3                | 4           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 4  | 4  | NORD | ANDINE  | 1          | Roberto Rossi | 4                | 5           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 5  | 5  | NORD | ANDINE  | 1          | Roberto Rossi | 5                | 5           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 6  | 6  | NORD | ANDINE  | 1          | Roberto Rossi | 6                | 5           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 7  | 7  | NORD | ANDINE  | 1          | Roberto Rossi | 7                | 5           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 8  | 8  | NORD | ANDINE  | 1          | Roberto Rossi | 8                | 5           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 9  | 9  | NORD | ANDINE  | 1          | Roberto Rossi | 9                | 5           | 2014/11/15  | 2015-01-01       |          |         |           |         |
|      | CASA 10 | 10 | NORD | ANDINE  | 1          | Roberto Rossi | 10               | 5           | 2014/11/15  | 2015-01-01       |          |         |           |         |

Numero case abitanti 225

Per ogni casa dell'elenco, sono disponibili numerose funzionalità.

- **Foto:** è possibile caricare una foto della casa.
- **Nome:** contiene il nome della casa.
- **Id:** identifica in modo univoco la casa.
- **Zona:** identifica in quale zona si trova la casa.
- **Morança:** identifica il nome della morança di appartenenza.
- **Id morança:** indica il numero delle case presenti nella morança.
- **Capo famiglia:** indica il nome del capo famiglia di riferimento.
- **Id capo famiglia:** identifica in modo univoco il capo famiglia.
- **N. abitanti:** indica il numero di abitanti presenti all'interno delle singole case.
- **Sulla mappa:** attraverso un link, è possibile visualizzare direttamente sulla mappa OSM dove si trova la casa in questione.
- **Data inizio validità:** indica in quale data è stata aggiunta la casa.

Vengono inoltre previsti diversi metodi per l'ordinamento dei dati in base alle esigenze dell'utente.

- **Zona:** attraverso un menù a tendina è possibile scegliere se mostrare tutte le case oppure solo quelle che fanno parte di una specifica zona.



- **Nome:** è possibile visualizzare le case in ordine alfabetico dalla A alla Z o viceversa.
- **Id:** è possibile ordinare le case in modo crescente o decrescente in base al numero che le identifica.



## 4.4 L'analisi delle specifiche software

Le specifiche tecniche di un software devono essere descritte in un idoneo documento. Per fare ciò, è necessario innanzitutto definire l'architettura generale del sistema.

### ■ L'architettura del sistema software

#### #techwords

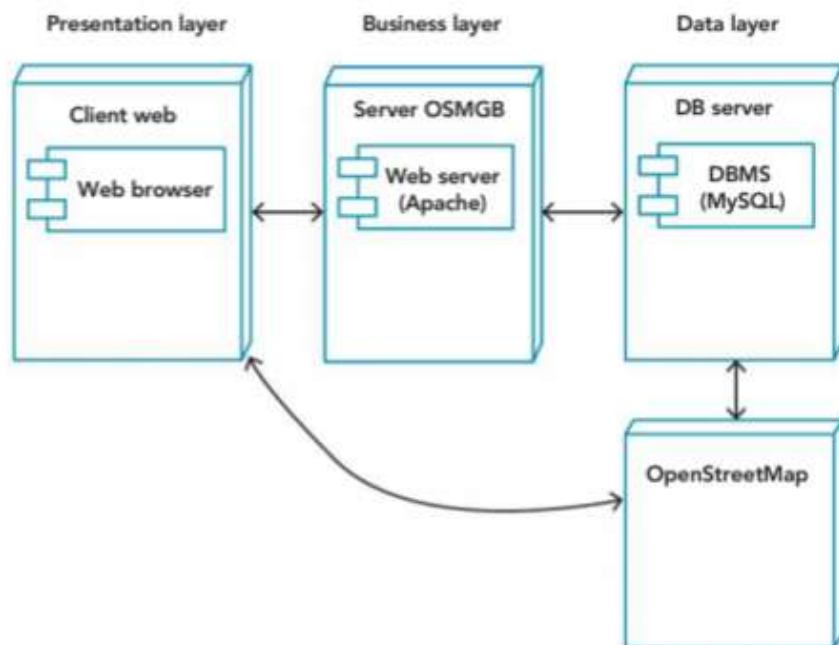
#### DIAGRAMMA DI DEPLOYMENT

È un diagramma statico previsto dal linguaggio UML per descrivere un sistema in termini di risorse hardware, dette nodi, di componenti software e di relazioni tra di essi.

Per descrivere l'architettura del sistema da realizzare è necessario dettagliare i componenti tecnologici mediante una rappresentazione grafica o altra specifica metodologia. Per esempio, ci si può servire di un **#diagramma di deployment** secondo il linguaggio UML.

#### CASO

Nel presente caso, come mostrato in figura, è impiegata una classica **architettura web a tre livelli**.



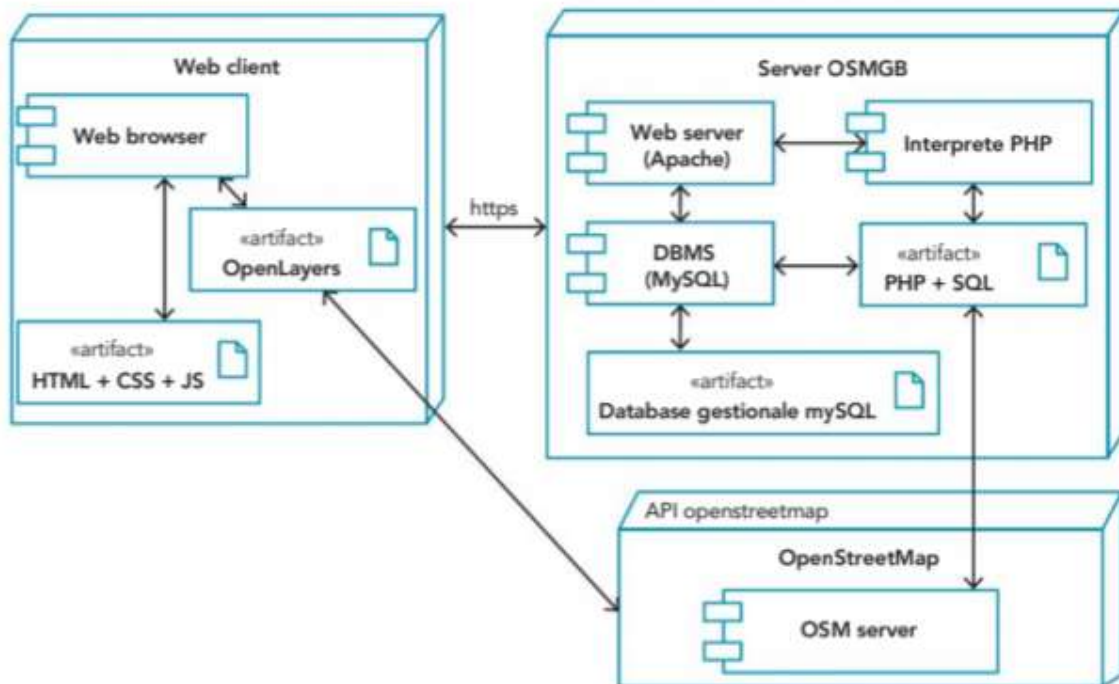
- **Client web:** ambiente di esecuzione dal lato client, in cui viene eseguito il browser, che visualizza pagine HTML, fogli CSS e JavaScript.
- **Server:** ambiente di esecuzione dal lato server, in cui viene eseguito il web server (Apache, in questo caso). L'interprete (PHP in questo caso) decodifica le pagine server in linguaggio PHP e produce pagine HTML in risposta al client. Nel codice PHP vengono effettuati gli accessi al database attraverso l'esecuzione di istruzioni SQL.
- **DB server:** ambiente di esecuzione che ospita il DBMS (DataBase Management System) MySQL e il database.

Occorre precisare che il livello server e il livello database server possono risiedere sulla stessa macchina fisica.

## 4.5 I componenti software e i linguaggi di programmazione

### CASO

Come visibile nello schema seguente, l'applicazione in questione è web client/server.



Sul nodo client, il componente software è il **web browser**, che ha il compito di visualizzare le pagine **HTML** con i relativi **fogli di stile CSS** ed effettuare i controlli con **JavaScript**. Il client effettua le richieste al nodo server via protocollo **HTTPS**. Il server, grazie al componente software **web server** (nel presente caso **Apache**), riceve le richieste dal client e, tramite **interprete PHP**, attiva gli script **PHP**. All'interno di questi ultimi vi sono le **query SQL** per l'accesso al **database mySQL**, che generano la risposta verso il client mediante pagine **HTML**.

L'accesso del client alla mappa **OpenStreetMap** è reso possibile dall'utilizzo di **OpenLayers**, una specifica libreria javascript. Dal lato server invece sono state utilizzate direttamente dal codice **PHP** alcune **#API**, per esempio per recuperare da **OSM** le coordinate di un nodo sulla mappa, o per recuperare il nodo che contiene una determinata coordinata spaziale.

Per l'interscambio dati tra applicazione gestionale e mappa è stato utilizzato il formato **geojson**, un formato basato su **#json**, contenente un array di punti, detto **feature**, con l'indicazione dei dati, tra cui latitudine e longitudine. Il file **geojson** contenente i dati delle case censite viene generato a runtime a partire dal database e rimane sempre sincronizzato con esso, a fronte delle modifiche effettuate sulla mappa.

### #techwords

#### API

Acronimo di Application Program Interface. In un programma informatico, è un insieme di procedure e protocolli per svolgere un dato compito.

#### JSON

Formato di interscambio per i dati, usato per la comunicazione tra applicazioni remote.

## 4.6 Gli ambienti operativi

### ■ I tipi di ambienti



FIGURA 2 Ambienti operativi

Oltre ai metodi di lavoro, è importante identificare e organizzare gli ambienti operativi sui quali installare il software.

Nel presente caso, trattandosi di un'applicazione web, è necessario installare il software su un server raggiungibile via rete. Normalmente si effettuano le prime **prove su un computer locale** (localhost), utilizzando tool come XAMPP che permettono l'utilizzo di un ambiente completo di web server Apache, DBMS MySQL e interprete PHP. Per una vera e propria installazione su un server dei propri programmi, esistono molti **servizi di hosting**, anche gratuiti (come per esempio [altervista.it](http://altervista.it)).

Di norma è quindi consigliabile l'utilizzo di tre ambienti, schematizzati in FIGURA 2.

- **Sviluppo** (ambiente localhost). È dedicato a ospitare l'applicazione per svolgere lo sviluppo di nuove funzionalità e i test locali.
- **Test**. È dedicato a ospitare l'applicazione per svolgere i test del sistema.
- **Produzione**. È destinato a ospitare l'applicazione finale a uso del cliente.

### ■ Gli ambienti condivisi

Un classico problema che si riscontra in un team di sviluppo software è la gestione delle versioni e il coordinamento e sincronizzazione delle attività di implementazione. Ogni membro del team deve poter aggiungere nuove funzionalità, apportare correzioni o tornare a una condizione precedente. Ciò comporta potenziali conflitti e la necessità di appositi strumenti, definiti **VCS** (Version Control Systems) e **DVCS** (Distributed Version Control Systems). I primi sono sistemi centralizzati e necessitano di un server remoto da cui caricare e scaricare i file. I secondi possono ricorrere o meno a un server remoto e possono operare anche in assenza di rete; tutte le modifiche vengono infatti registrate localmente e possono essere sincronizzate in un secondo momento. A oggi il DVCS più diffuso è **Git**: si tratta di un DVCS open source, creato da Linus Torvalds nel 2005 per gestire il versioning del sistema operativo Linux.

**GitHub**, invece, è un servizio di hosting di repository Git basato su cloud. Rispetto a Git rende più facile la gestione dei progetti collegati ai repository: ha infatti un'interfaccia semplice da utilizzare, mentre Git richiede l'utilizzo della linea di comando. Chiunque può iscriversi e ospitare gratuitamente un repository pubblico, cioè il cui codice sorgente può essere visto e scaricato da tutti. Questo rende GitHub particolarmente popolare tra i progetti open source.

## 4.7 La progettazione del database

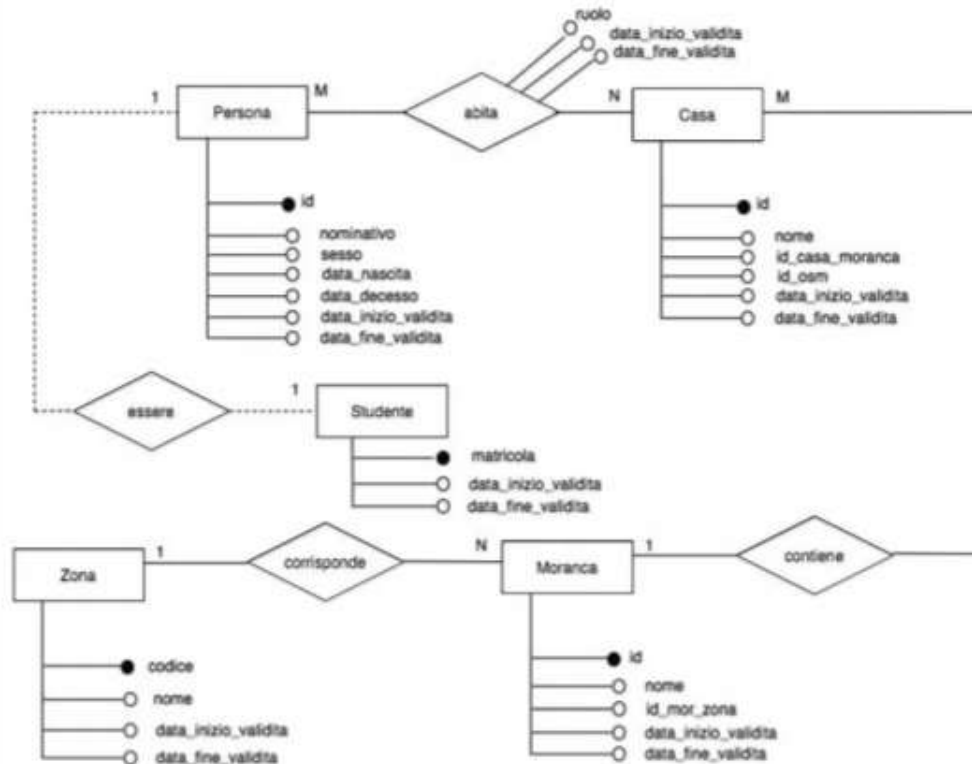
La progettazione del database è di fondamentale importanza. Nel caso di utilizzo di un database di natura relazionale, a seguito della fase di analisi dei dati da gestire dall'applicazione, occorre definire uno schema concettuale, secondo il classico **modello E/R** (entità/relazioni), e il corrispettivo **schema logico del database**.

Come abbiamo imparato nel corso di informatica, seguiamo le regole di progettazione in modo che il database sia in terza forma normale.

Le tabelle risultanti devono essere descritte e si devono indicare i singoli campi e i vincoli (chiavi primarie, chiavi esterne, ecc.).

## Lo schema concettuale del database (modello E/R)

### Caso



Vediamo in dettaglio le entità coinvolte.

L'entità **Persona** rappresenta le persone, ovvero gli abitanti del villaggio:

- **id** è l'identificativo numerico associato a una persona;
- **nominativo** è il nominativo della persona;
- **sesso** è il sesso della persona (valori possibili 'M' o 'F');
- **data\_nascita** è la data di nascita della persona;
- **data\_inizio\_validita** e **data\_fine\_validita** sono relative alle date di validità del dato;
- **data\_decesso** è l'eventuale data di decesso della persona.

L'entità **Casa** rappresenta le case dove vivono le persone:

- **id** è l'identificativo numerico della casa;
- **nome** è il nome della casa, di norma corrisponde al nome del capofamiglia che la occupa;
- **id\_casa\_moranca** è l'identificativo della casa all'interno della moranca;
- **id\_osm** è l'identificativo su OpenStreetMap dell'oggetto associato alla casa, a fronte della mappatura della casa stessa;
- **data\_inizio\_validita** è la data di inizio di validità del dato;
- **data\_fine\_validita** è la data di fine di validità del dato.



L'entità **Moranca** rappresenta le moranças, ovvero i raggruppamenti di case:

- **id** è l'identificativo numerico della morança;
- **nome** è il nome della morança;
- **id\_mor\_zona** è l'identificativo della morança all'interno della zona;
- **data\_inizio\_validita** è la data di inizio di validità del dato;
- **data\_fine\_validita** è la data di fine di validità del dato.

L'entità **Zona** rappresenta le zone del villaggio di N'Tchangue, ove si trovano le moranças:

- **codice** è il codice identificativo della zona del villaggio ('N' è nord, 'S' è sud, 'O' è ovest);
- **nome** è il nome della zona;
- **data\_inizio\_validita** è la data di inizio di validità del dato;
- **data\_fine\_validita** è la data di fine di validità del dato.

L'entità **Studiante** rappresenta gli studenti della scuola del villaggio di N'Tchangue:

- **matricola** è il codice identificativo dello studente;
- **data\_inizio\_validita** è la data di inizio di validità del dato;
- **data\_fine\_validita** è la data di fine di validità del dato.

Tra le entità descritte sussistono vari tipi di relazioni.

- **abita (1, N).**  
Una persona abita in una casa.  
In una casa possono abitare da 1 a N persone.  
Ogni abitante ha un ruolo di appartenenza all'interno della casa.
- **contiene (1, N).**  
Una morança può contenere da 1 a N case.  
Una casa deve essere contenuta in una e una sola morança.
- **corrisponde (1, N).**  
A una zona possono corrispondere da 1 a N moranças.  
Una morança deve corrispondere a una sola zona.
- **essere (1, 1).**  
Una persona può essere uno studente.  
Uno studente deve essere una persona.

## ■ Lo schema logico del database

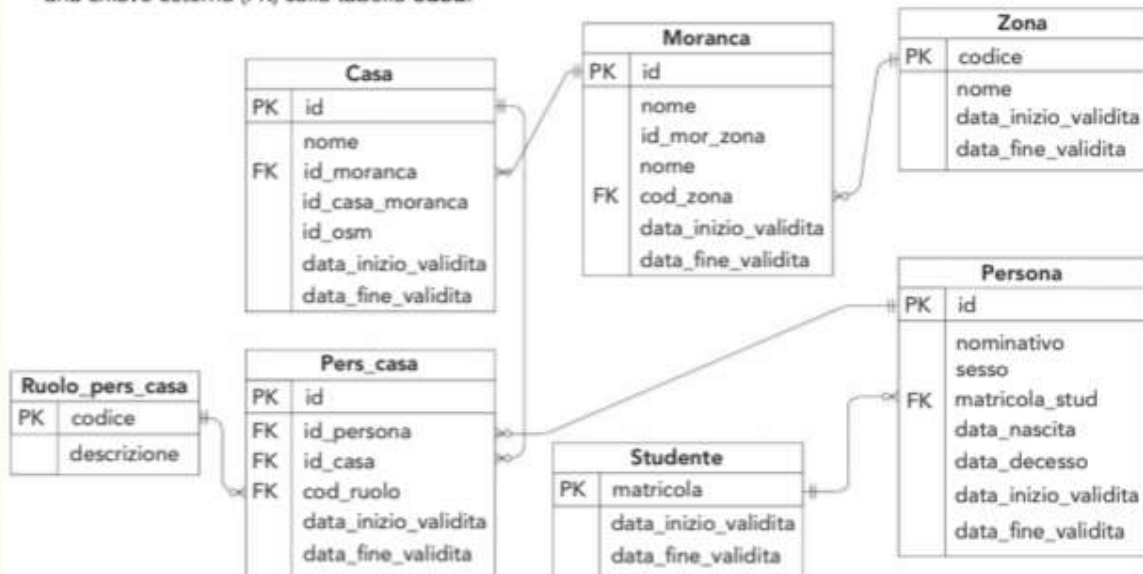
Dallo schema E/R è possibile derivare lo schema logico del database seguendo una serie di **regole di derivazione**.

### caso

- Le entità diventano delle tabelle:
 

|                 |                    |
|-----------------|--------------------|
| Entità Zona     | → Tabella Zona     |
| Entità Moranca  | → Tabella Moranca  |
| Entità Casa     | → Tabella Casa     |
| Entità Persona  | → Tabella Persona  |
| Entità Studente | → Tabella Studente |

- La relazione **corrisponde** tra **Zona** e **Moranca**, essendo di cardinalità 1, N, diventa una chiave esterna (Foreign Key, FK) sulla tabella **Moranca**.
- La relazione **abita** tra **Persona** e **Casa**, essendo di cardinalità M, N, diventa una tabella associativa denominata **Pers\_casa**.
- La relazione **essere** tra **Persona** e **Studiante**, essendo di cardinalità 1, 1, diventa una chiave esterna (FK) sulla tabella **Studiante** (poteva anche essere scelta una FK sulla tabella **Persona**).
- La relazione **contiene** tra **Casa** e **Moranca**, essendo di cardinalità 1, N, diventa una chiave esterna (FK) sulla tabella **Casa**.



Di seguito sono descritte alcune tabelle significative dello schema logico del DB: sono state omesse per semplicità le tabelle con i dati storici di persone, case e morançs, e le tabelle di gestione degli account e dei trace.

#### Casa

Nome tabella: **Casa**

Descrizione: tabella che contiene i dati sulle case del villaggio.

Attributi:

| Nome            | Descrizione                          | Tipo       | Vincoli         |
|-----------------|--------------------------------------|------------|-----------------|
| id              | identificativo della casa            | int        | PK              |
| nome            | nome della casa                      | varchar 50 |                 |
| id_moranca      | identificativo della morançs         | int        | FK (Moranca.id) |
| id_casa_moranca | identificativo della casa nell'Excel | int        |                 |
| id_osm          | identificativo della casa su OSM     | int        |                 |
| lat             | latitudine                           | float      |                 |
| lon             | longitudine                          | float      |                 |
| data_inizio_val | data di inizio di validità del dato  | date       |                 |
| data_fine_val   | data di fine di validità del dato    | date       |                 |

**Persona**Nome tabella: **Persona**

Descrizione: tabella contenente i dati degli abitanti di N'Tchague.

Attributi:

| Nome            | Descrizione                         | Tipo       | Vincoli                 |
|-----------------|-------------------------------------|------------|-------------------------|
| id              | identificativo della persona        | int        | PK                      |
| nome            | nominativo della persona            | varchar 50 |                         |
| sezzo           | 'M' o 'F'                           | char 1     |                         |
| matricola_stud  | matricola, se studente              | varchar 50 | FK (Studente.matricola) |
| data_nascita    | data di nascita                     | date       |                         |
| data_inizio_val | data di inizio di validità del dato | date       |                         |
| data_fine_val   | data di fine di validità del dato   | date       |                         |

**Pers\_casa**Nome tabella: **Pers\_casa**

Descrizione: tabella contenente le associazioni tra le persone e le rispettive abitazioni.

Attributi:

| Nome            | Descrizione                         | Tipo       | Vincoli                     |
|-----------------|-------------------------------------|------------|-----------------------------|
| id              | identificativo                      | int        | PK<br>autoincrement         |
| id_pers         | identificativo della persona        | int        | FK (Persone.id)             |
| id_casa         | identificativo della casa           | int        | FK (casa.id)                |
| cod_ruolo       | ruolo della persona nella famiglia  | varchar 10 | FK (Ruolo_pers_casa.codice) |
| data_inizio_val | data di inizio di validità del dato | date       |                             |
| data_fine_val   | data di fine di validità del dato   | date       |                             |

■ **Strumenti per lo sviluppo del software**

Per la scrittura del codice sorgente, si possono utilizzare strumenti cosiddetti IDE (Integrated Developer Environment), che permettono di gestire le fasi di sviluppo in maniera efficiente e integrata.

Si possono impiegare tool differenti in base alla tipologia di linguaggio o alle preferenze del singolo sviluppatore. Tra i più diffusi ricordiamo:

- NetBeans;
- IntelliJ;
- Xcode;
- Visual Studio Code.

Per lo sviluppo delle interrogazioni SQL si utilizza l'ambiente integrato XAMPP, che fornisce l'accesso al DBMS MySQL sia in modalità comando sia con il tool PHPMyAdmin.

## 4.8 Le metriche del software

Un software è costituito da **codice sorgente** scritto in uno o più linguaggi di programmazione, normalmente compilati o interpretati, che viene poi eseguito su un server e agisce sui dati. Essendo immateriale, non è di per sé visibile se non attraverso modelli o formalismi simbolici e astrazioni.

Per descrivere i suoi componenti si ricorre a linguaggi visuali di modellazione, come per esempio i flow chart (diagrammi di flusso) oppure i diagrammi UML.

Fin dai suoi albori, l'ingegneria informatica ha cercato di definire delle **metriche** in grado di misurare il software, in modo da poterne stimare in modo **oggettivo** le dimensioni e di conseguenza i costi, i tempi e le risorse necessarie per realizzarlo.

La misura del software, nel ciclo di vita del software, può essere svolta sia come stima preventiva sia come verifica a consuntivo, confrontabile con altre applicazioni similari. Ogni azienda ICT dovrebbe conservare la storia dei prodotti realizzati nel corso del tempo per aumentare il grado di precisione delle stime di misura.

### ■ La LOC (Line Of Code)

Le prime tecniche di misurazione risalgono agli anni Sessanta del secolo scorso e si basano sul **conteggio del numero di righe** di codice sorgente (in inglese *line of code*): più righe vengono scritte dai programmatori, maggiore risulta la complessità del software e quindi i relativi valore e costo.

### ■ La complessità ciclomatica

Nel caso di applicazioni strettamente legate al calcolo matematico e algoritmico, è possibile estrarre dalla struttura del codice un grafo comprendente un certo numero di cammini o diramazioni. Questi corrispondono alla presenza nel codice sorgente di istruzioni condizionali (**if**, **switch case**) e ripetitive (**for**, **while**, **do while**). In base al numero di possibili cammini, viene identificato un grado di complessità che può rappresentare, sempre che il codice sia ben scritto, l'effettiva misura del software in oggetto. Questo sistema di misura viene comunemente detto **calcolo ciclomatico**.

### ■ Le misurazioni funzionali (IFPUG)

Alla fine degli anni Settanta del secolo scorso, un gruppo di ricercatori dell'IBM, guidato da Allan J. Albrecht, definì una metrica che, a partire dai requisiti funzionali di un'applicazione (ossia la definizione di che cosa fa e non di come è realizzata), permette di ottenere una misura del software in **function point** (FP), letteralmente punti funzione. Rispetto alle metriche basate sulle LOC, la misura funzionale permette di ottenere una stima prima che il software sia sviluppato.

La metodologia è gestita dall'**IFPUG** (International Function Point User Group), che ne definisce le regole in un manuale del conteggio e ne governa la certificazione.

Per effettuare una misurazione funzionale è necessario disporre di un **modello logico dei dati** e di uno schema delle viste che si prevedono di realizzare per l'interfaccia utente. La metodologia IFPUG individua due tipologie di funzioni:

- le **funzioni dati**, identificate dalle entità logiche su cui il sistema opera;
- le **funzioni transazioni**, identificate da processi elementari di inserimento, modifica, cancellazione e visualizzazione delle entità, che descrivono le funzionalità offerte agli utenti.



### Le funzioni dati

Le **funzioni dati** (o entità logiche) di un'applicazione che si intende misurare sono le informazioni riconoscibili dall'utente (sia esso un operatore umano oppure un'applicazione), che devono costituire un insieme coerente, indipendente dalle altre entità logiche.

Tali informazioni possono essere:

- interne, dette **ILF** (Internal Logical File), se sono gestite dall'applicazione che si sta esaminando;
- esterne, dette **EIF** (External Interface File), se sono gestite da un'altra applicazione e possono essere lette dall'applicazione in esame.

A ogni funzione dati è assegnata una complessità in base:

- al numero di entità fisiche che compongono la relazione (denominate **RET**, Record Element Type);
- al numero di attributi (denominati **DET**, Data Element Type).

A ogni complessità è assegnato un numero ben preciso di function point.

Nella **TABELLA 1** si trovano le formule per calcolare la complessità della funzione dati.

#### ILF/EIF - Matrice di complessità

|         | 1-19 DET | 20-50 DET | > 50 DET |
|---------|----------|-----------|----------|
| 1 RET   | Bassa    | Bassa     | Media    |
| 2-5 RET | Bassa    | Media     | Alta     |
| > 5 RET | Media    | Alta      | Alta     |

**TABELLA 1** Formule per calcolare la complessità della funzione dati

#### Tabelle di Conversione in FP: ILF

| Complessità | FP |
|-------------|----|
| Bassa       | 7  |
| Media       | 10 |
| Alta        | 15 |

#### EIF

| Complessità | FP |
|-------------|----|
| Bassa       | 5  |
| Media       | 7  |
| Alta        | 10 |

### Le funzioni transazioni

Le **funzioni transazioni** sono le operazioni di gestione dei dati. Si tratta tipicamente delle cosiddette **CRUD operation**: inserimenti, visualizzazioni, modifiche, cancellazioni (CRUD deriva dall'inglese Create, Read, Update, Delete), ma anche ricerche, ricezione di dati da altre applicazioni, invio di dati all'esterno.

Ognuna di queste azioni sui dati viene eseguita all'interno di un **processo elementare**, cioè una serie di azioni al termine delle quali il sistema si trova in una situazione di coerenza funzionale. Le funzioni transazioni individuano i seguenti processi elementari:

- **EI** (External Input), se hanno lo scopo di gestire i dati attraverso un input (per esempio funzioni di inserimento, modifica o cancellazione);
- **EQ** (External Inquiry), se hanno lo scopo di visualizzare i dati (per esempio risultati di una query, una ricerca, report o export su file senza elaborazione);
- **EO** (External Output), se hanno come scopo l'invio di dati all'esterno a seguito di una loro elaborazione (per esempio statistiche, report di conteggio, export su file con elaborazione).

A ogni funzione transazione è assegnata una complessità sulla base:

- del numero di entità coinvolte in lettura e scrittura (dette **FTR**, File Type Referenced);
- del numero di attributi coinvolti (denominati **DET**, Data Element Type), che corrispondono alle più piccole unità di informazione logica riconosciute dall'utente.

Non necessariamente gli attributi sono collegati a un solo campo di una videata. Per esempio, nel caso di un indirizzo, il DET può essere composto da più campi: indirizzo, numero civico, CAP, città, provincia. Nel caso in cui vi sia un insieme di righe ripetute, si calcola il numero di DET presenti nella prima riga.

Nella **TABELLA 2** si trovano le formule per calcolare la complessità della funzione transazione. A ogni complessità è assegnato un numero preciso di function point.

#### EI

|         | 1-4 DET | 5-15 DET | > 15 DET |
|---------|---------|----------|----------|
| 0-1 FTR | Bassa   | Bassa    | Media    |
| 2 FTR   | Bassa   | Media    | Alta     |
| > 2 FTR | Media   | Alta     | Alta     |

**TABELLA 2** Formule per calcolare la complessità della funzione transazione

#### EO/EQ

|         | 1-5 DET | 6-19 DET | > 19 DET |
|---------|---------|----------|----------|
| 0-1 FTR | Bassa   | Bassa    | Media    |
| 2-3 FTR | Bassa   | Media    | Alta     |
| > 3 FTR | Media   | Alta     | Alta     |

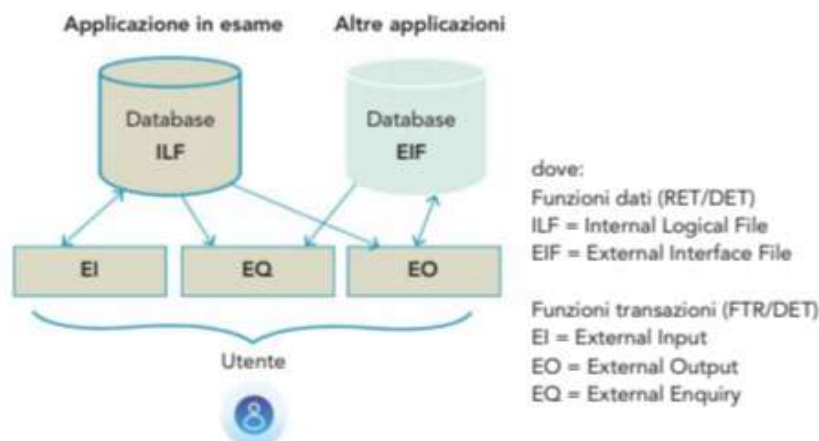
#### EI/EQ

| Complessità | FP |
|-------------|----|
| Bassa       | 3  |
| Media       | 4  |
| Alta        | 6  |

#### EO

| Complessità | FP |
|-------------|----|
| Bassa       | 4  |
| Media       | 5  |
| Alta        | 7  |

Nella **FIGURA 3** è schematizzata in maniera riassuntiva la metodologia IFPUG.



**FIGURA 3** La metodologia IFPUG

### ■ Le altre metodologie

A seguito dello sviluppo della metodologia IFPUG (attualmente ancora la più diffusa), sono nate altre metodologie (Nesma, Mark II, COSMIC) che con regole differenti misurano la dimensione funzionale di un'applicazione, sempre sulla base della movimentazione delle informazioni gestite.

Alla dimensione funzionale possono essere associate altre informazioni, prime fra tutte:

- la **produttività di sviluppo**, cioè quanti function point sono sviluppati nell'unità di tempo;
- la **difettosità**, cioè quanti difetti dell'applicazione si riscontrano per ogni function point.

Se un'azienda ICT dispone di uno storico delle misure rilevate nei progetti pregressi, è possibile trarre informazioni importanti dai dati e, per esempio, definire modelli di stima dell'impegno nel momento in cui si misurano i requisiti funzionali di un progetto ancora da realizzare. Questi dati sono essenziali quando occorre creare un preventivo per quotare economicamente lo sviluppo di un software.

### ■ Misure di qualità

Anche la qualità del software è quanto mai importante e andrebbe sempre misurata. Nel contesto software è stato definito dall'ISO il paradigma **SQuaRE** (Software product Quality Requirement and Evaluation) che propone una serie di caratteristiche e di relativi indicatori di qualità. In particolare:

- norme ISO 25010-25023 per la qualità del prodotto;
- norme ISO 25011-25022 per la qualità del prodotto in uso (servizio);
- norme ISO 25012-25024 per la qualità dei dati.

Gli indicatori di qualità di un prodotto software (per esempio la manutenibilità, l'usabilità, la sicurezza, ecc.) possono essere ottenuti attraverso strumenti di mercato (alcuni free, altri con licenza, come per esempio Plandek, [www.plandek.com](http://www.plandek.com)).



### ■ Le misure nei contratti

Le metriche descritte nei Paragrafi precedenti possono essere utilizzate anche per la stipula dei contratti di fornitura del software tra un committente e un fornitore.

In particolare nell'ambito della Pubblica amministrazione vengono bandite gare d'appalto per un certo numero di function point: è stabilito un prezzo base di partenza per function point e, tramite un algoritmo che prende in considerazione altre caratteristiche (per esempio l'offerta tecnica, l'esperienza del fornitore, ecc.), viene aggiudicata la gara al miglior partecipante. Il fornitore aggiudicatario dovrà misurare, progetto per progetto, i function point e rispettare le misure di qualità definite nel contratto. Le metodologie di misurazione sono un utile strumento di supporto al project manager ma, come per le altre tecniche, non possono sostituirsi alla sua esperienza e competenza.



Leggi  
l'**approfondimento**  
online

## 4.9 La misurazione del progetto anagrafe e mappa del villaggio di N'Tchangue

Una volta che si hanno a disposizione il modello logico dei dati, i diversi casi d'uso e le videate dell'applicazione, è possibile effettuare una misurazione del progetto Anagrafe web e Mappa OpenStreetMap del villaggio di N'Tchangue usando la **metodologia IFPUG**. Come abbiamo visto nel Paragrafo precedente, questa metodologia individua le **funzioni dati** (le entità logiche) e le **funzioni transazioni** (i processi di inserimento, modifica, cancellazione e visualizzazione delle entità).

### CASO

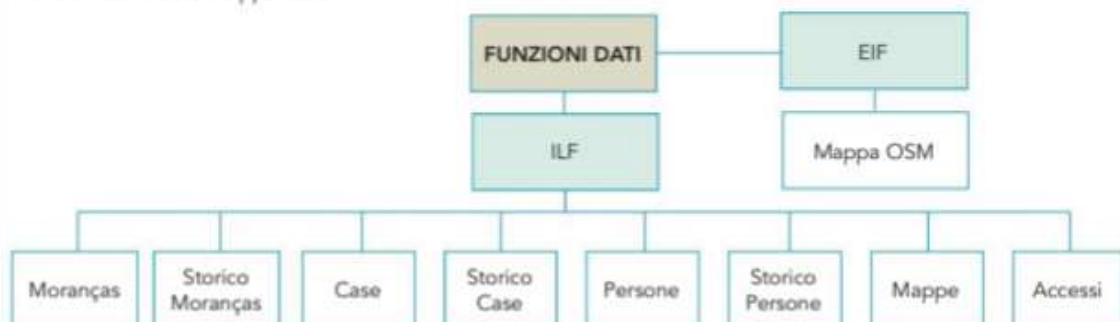
Di seguito sono esposti i calcoli effettuati sull'intero progetto.

#### Funzioni dati

Le funzioni dati devono considerare tutte le entità logiche. Si identificano i seguenti ILF:

- Moranças;
- Storico Moranças;
- Case;
- Storico Case;
- Persone;
- Storico Persone;
- Mappe;
- Accessi.

Si identifica un EIF: Mappa OSM.



#### Funzioni transazioni

Le funzioni transazioni dell'intera applicazione devono considerare tutte le funzionalità visibili dal punto di vista utente. Si considerano funzioni transazioni (EI, EQ, EO):

- Gestione Moranças;
- Gestione Case;
- Gestione Persone;
- Gestione Storico Moranças;
- Case e Persone;
- Accessi al sistema;
- Integrazione con Mappa OSM.





**Misurazione della Gestione Moranças**

Vediamo ora nel dettaglio la misura delle funzionalità per la gestione delle moranças.

- **Funzioni dati.** Le moranças sono gruppi di case. Esiste quindi una relazione 1, N tra la tabella Morança e la tabella Casa. Una casa non può esistere se non all'interno di una morança. Come rappresentato nella seconda figura, si identificano quindi le moranças come Internal Logical File (ILF) composti da 2 RET (morança e casa) e 19 DET (la somma degli attributi delle due tabelle).



Con 2 RET e 19 DET (vedi **TABELLA 2** pagina 217) si ha una complessità bassa, da cui risultano 7 FP.

| Funzione dati |      |     |     |             |    |
|---------------|------|-----|-----|-------------|----|
| Nome          | Tipo | RET | DET | Complessità | FP |
| Morança       | ILF  | 2   | 19  | Bassa       | 7  |

| ILF              | RETs | DETs | Complessità | FP | Note             |
|------------------|------|------|-------------|----|------------------|
| Moranças         | 2    | 19   | Bassa       | 7  | morança, case    |
| Storico moranças | 1    | 7    | Bassa       | 7  |                  |
| Case             | 2    | 14   | Bassa       | 7  | case, persone    |
| Storico case     | 1    | 7    | Bassa       | 7  |                  |
| Mappa (case)     | 1    | 3    | Bassa       | 7  | 1 livello (case) |
| Persone          | 2    | 15   | Bassa       | 7  | persone, case    |
| Storico persone  | 1    | 9    | Bassa       | 7  |                  |
| Accessi          | 1    | 5    | Bassa       | 7  |                  |
| Sub Totale       |      |      |             | 56 |                  |

| EIF                 | RETs | DETs | Complessità | FP | Note          |
|---------------------|------|------|-------------|----|---------------|
| Mappa (servizi API) | 1    | 3    | Bassa       | 5  | morança, case |
| Mappa GIS           | 1    | 2    | Bassa       | 5  |               |
| Sub Totale          |      |      |             | 10 |               |

La componente relativa alla Funzione dati misura quindi  $56 + 10 = 66$  FP.

- **Funzioni transazioni.** Prendendo come esempio i casi d'uso relativi alla Gestione Moranças, si possono identificare i processi elementari dettagliati nella tabella presentata nella pagina seguente.

| Processi elementari (Gestione Moranças) |      |
|---|------|
| Nome                                    | Tipo |
| Visualizza elenco moranças              | EQ   |
| Cerca moranças                          | EO   |
| Cerca moranças per zona                 | EO   |
| Inserimento morança                     | EI   |
| Elimina morança                         | EI   |
| Modifica morança                        | EI   |
| Export su Excel                         | EQ   |
| Visualizza case                         | EQ   |
| Visualizza storico                      | EO   |
| Visualizza foto                         | EQ   |
| Visualizza su mappa                     | EQ   |

**Misura Progetto OSM**  
Processi elementari gestione della Moranças

| ID | Nome    | Area | Proj. area | Numero case | Numero edifici | Area mappa | Data creazione | Modifica | Elimina | Casa | Storico |
|----|---------|------|------------|-------------|----------------|------------|----------------|----------|---------|------|---------|
| 1  | MORANCA | 01   | 001        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 2  | MORANCA | 01   | 002        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 3  | MORANCA | 01   | 003        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 4  | MORANCA | 01   | 004        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 5  | MORANCA | 01   | 005        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 6  | MORANCA | 01   | 006        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 7  | MORANCA | 01   | 007        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 8  | MORANCA | 01   | 008        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 9  | MORANCA | 01   | 009        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |
| 10 | MORANCA | 01   | 010        | 24          | 3              | 41         | 2008-12-08     |          |         |      |         |

Se ora ci soffermiamo, per esempio, sulla misura della funzionalità "Modifica morança", nel processo elementare di modifica di una morança si identificano 2 FTR (Morança e Storico morança) con 5 DET (i tre campi modificabili e un pulsante per l'azione conferma o annulla e un eventuale messaggio di errore o informazione).

**N'Tchangué AnagniniWEB**

Modifica morança: MORANCA 01-001 area: 0001

Nome morança: MORANCA

Seleziona zona: 0001

Area mappa:

**Modifica**

Modifica la foto della morança.

Seleziona una foto da caricare:

**Carica foto**

Attualmente non è presente alcuna foto.

[Torna a selezione morança](#)

Di conseguenza, "Modifica morança" risulta essere un External Input con 2 FTR e 5 DET. Questo equivale a una complessità media, ovvero 4 FP.

| Funzioni transazioni |      |     |     |             |    |
|----------------------|------|-----|-----|-------------|----|
| Nome                 | Tipo | FTR | DET | Complessità | FP |
| Modifica morança     | EI   | 2   | 5   | Media       | 4  |

Allo stesso modo si possono calcolare gli altri processi elementari. Adottando la stessa metodologia IFPUG su tutti i casi d'uso identificati dall'applicazione, grazie all'utilizzo di un foglio di calcolo opportunamente impostato, si può effettuare il **calcolo complessivo**, riportato parzialmente nel prospetto che segue.

Come si vede, effettuando il conteggio su tutte le funzioni dati e tutte le funzioni transazioni, il progetto risulta misurare 310 FP secondo il sistema di misurazione IFPUG.

| Oggetto da misurare                | Funzione (EI,EO,EQ) | FTRs |       | DETs |       | Compl. Totale | FP |    |    |      |
|------------------------------------|---------------------|------|-------|------|-------|---------------|----|----|----|------|
|                                    |                     | EI   | EO/EQ | EI   | EO/EQ |               | EI | EO | EQ | Tot. |
| Login                              | EI                  | 2    |       | 4    |       | Bassa         | 3  |    |    | 3    |
| Gestione Moranças                  |                     |      |       |      |       |               |    |    |    |      |
| Ricerca                            | EO                  |      | 2     |      | 11    | Media         |    | 5  |    | 5    |
| Ricerca per zona                   | EO                  |      | 2     |      | 11    | Media         |    | 5  |    | 5    |
| Elenco                             | EQ                  |      | 1     |      | 11    | Bassa         |    |    | 3  | 3    |
| Visualizza foto                    | EQ                  |      | 1     |      | 2     | Bassa         |    |    | 3  | 3    |
| Visualizza storico                 | EQ                  |      | 2     |      | 8     | Media         |    |    | 4  | 4    |
| Inserimento                        | EI                  | 1    |       | 5    |       | Bassa         | 3  |    |    | 3    |
| Modifica                           | EI                  | 2    |       | 5    |       | Media         | 4  |    |    | 4    |
| Elimina                            | EI                  | 2    |       | 2    |       | Bassa         | 3  |    |    | 3    |
| Visualizza mappa                   | EQ                  |      | 3     |      | 2     | Bassa         |    |    | 3  | 3    |
| Case                               | EQ                  |      | 1     |      | 11    | Bassa         |    |    | 3  | 3    |
| Storia delle moranças (modificate) | EO                  |      | 1     |      | 10    | Bassa         |    | 4  |    | 4    |
| Storia delle moranças (eliminate)  | EO                  |      | 1     |      | 10    | Bassa         |    | 4  |    | 4    |
| Storia delle moranças (tutte)      | EO                  |      | 1     |      | 10    | Bassa         |    | 4  |    | 4    |
| Export su Excel                    | EQ                  |      | 2     |      | 11    | Media         |    |    | 4  | 4    |
| Gestione Case                      |                     |      |       |      |       |               |    |    |    |      |
| Ricerca                            | EO                  |      | 2     |      | 13    | Media         |    | 5  |    | 5    |
| Elenco                             | EQ                  |      | 1     |      | 11    | Bassa         |    |    | 3  | 3    |
| Visualizza foto                    | EQ                  |      | 1     |      | 2     | Bassa         |    |    | 3  | 3    |
| Visualizza storico                 | EQ                  |      | 2     |      | 10    | Media         |    |    | 4  | 4    |
| Inserimento                        | EI                  | 1    |       | 5    |       | Bassa         | 3  |    |    | 3    |
| Modifica                           | EI                  | 2    |       | 5    |       | Media         | 4  |    |    | 4    |
| Elimina                            | EI                  | 2    |       | 2    |       | Bassa         | 3  |    |    | 3    |
| Persone                            | EQ                  |      | 1     |      | 6     | Bassa         |    |    | 3  | 3    |
| Visualizza mappa                   | EQ                  |      | 2     |      | 2     | Bassa         |    |    | 3  | 3    |
| Storia delle case (modificate)     | EO                  |      | 1     |      | 10    | Bassa         |    | 4  |    | 4    |
| Storia delle case (eliminate)      | EO                  |      | 1     |      | 10    | Bassa         |    | 4  |    | 4    |
| Storia delle case (tutte)          | EO                  |      | 1     |      | 10    | Bassa         |    | 4  |    | 4    |
| Export su Excel                    | EQ                  |      | 2     |      | 13    | Media         |    |    | 4  | 4    |

|                       |                                     |    |   |   |    |    |            |    |     |    |     |
|-----------------------|-------------------------------------|----|---|---|----|----|------------|----|-----|----|-----|
| Gestione persone      |                                     |    |   |   |    |    |            |    |     |    |     |
|                       | Ricerca                             | EO |   | 2 |    | 14 | Media      |    | 5   |    | 5   |
|                       | Elenco (tutti)                      | EQ |   | 1 |    | 11 | Bassa      |    |     | 3  | 3   |
|                       | Elenco (viventi)                    | EQ |   | 1 |    | 11 | Bassa      |    |     | 3  | 3   |
|                       | Elenco (deceduti)                   | EQ |   | 1 |    | 11 | Bassa      |    |     | 3  | 3   |
|                       | Visualizza storico                  | EQ |   | 2 |    | 10 | Media      |    |     | 4  | 4   |
|                       | Inserimento                         | EI | 1 |   | 5  |    | Bassa      | 3  |     |    | 3   |
|                       | Modifica                            | EI | 2 |   | 5  |    | Media      | 4  |     |    | 4   |
|                       | Elimina                             | EI | 2 |   | 2  |    | Bassa      | 3  |     |    | 3   |
|                       | Case                                | EQ |   | 1 |    | 4  | Bassa      |    |     | 3  | 3   |
|                       | Visualizza mappa                    | EQ |   | 2 |    | 2  | Bassa      |    |     | 3  | 3   |
|                       | Storia delle persone (modificate)   | EO |   | 1 |    | 13 | Bassa      |    | 4   |    | 4   |
|                       | Storia delle persone (eliminate)    | EO |   | 1 |    | 13 | Bassa      |    | 4   |    | 4   |
|                       | Storia delle persone (tutte)        | EO |   | 1 |    | 13 | Bassa      |    | 4   |    | 4   |
|                       | Export su Excel                     | EQ |   | 2 |    | 13 | Media      |    |     | 4  | 4   |
| Mappa Case            |                                     |    |   |   |    |    |            |    |     |    |     |
|                       | Visualizza case su mappa            | EQ |   | 3 |    | 2  | Bassa      |    |     | 3  | 3   |
|                       | Visualizza dati su mappa            | EQ |   | 4 |    | 9  | Alta       |    |     | 6  | 6   |
|                       | Modifica                            | EI | 2 |   | 11 |    | Media      | 4  |     |    | 4   |
| Statistiche           |                                     |    |   |   |    |    |            |    |     |    |     |
|                       | Percentuali maschi e femmine        | EO |   | 1 |    | 4  | Bassa      |    |     | 4  | 4   |
|                       | Percentuali maggiorenni e minorenni | EO |   | 1 |    | 4  | Bassa      |    | 4   |    | 4   |
|                       | Persone per fasce di età            | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
|                       | Donne fertili                       | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
| Dettaglio Statistiche |                                     |    |   |   |    |    |            |    |     |    |     |
|                       | Statistiche per anno                | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
|                       | Statistiche decessi                 | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
|                       | Statistiche complessive             | EO |   | 1 |    | 4  | Bassa      |    | 4   |    | 4   |
| Statistiche per zona  |                                     |    |   |   |    |    |            |    |     |    |     |
|                       | Maschi e femmine                    | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
|                       | Maggiorenni                         | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
|                       | Fertili                             | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
|                       | Fasce                               | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
|                       | Numero di persone                   | EO |   | 1 |    | 3  | Bassa      |    | 4   |    | 4   |
| Utenti                |                                     |    |   |   |    |    |            |    |     |    |     |
|                       | Visualizza accessi (tutti)          | EO |   | 2 |    | 5  | Bassa      |    | 4   |    | 4   |
|                       | Visualizza accessi (falliti)        | EO |   | 2 |    | 5  | Bassa      |    | 4   |    | 4   |
|                       | Visualizza accessi (riusciti)       | EO |   | 2 |    | 5  | Bassa      |    | 4   |    | 4   |
|                       | Elenco                              | EO |   | 1 |    | 4  | Bassa      |    | 4   |    | 4   |
|                       | Inserisci                           | EI | 1 |   | 6  |    | Bassa      | 3  |     |    | 3   |
|                       | Elimina                             | EI | 1 |   | 2  |    | Bassa      | 3  |     |    | 3   |
| Admin                 |                                     |    |   |   |    |    |            |    |     |    |     |
|                       | Area personale                      | EQ |   | 1 |    | 4  | Bassa      |    |     | 3  | 3   |
|                       | Cambia password                     | EI | 1 |   | 3  |    | Bassa      | 3  |     |    | 3   |
|                       | Esci                                | EI | 1 |   | 2  |    | Bassa      | 3  |     |    | 3   |
|                       |                                     |    |   |   |    |    | Sub Totale | 49 | 120 | 75 | 244 |

Ai 244 così calcolati dobbiamo sommare i  $56 + 10$  già calcolati a p. 220.

|                                 |  |
|---------------------------------|--|
| Function Point Non Pesati - UFP | 310 (66 per i dati e 244 per le transazioni) |
|---------------------------------|--|

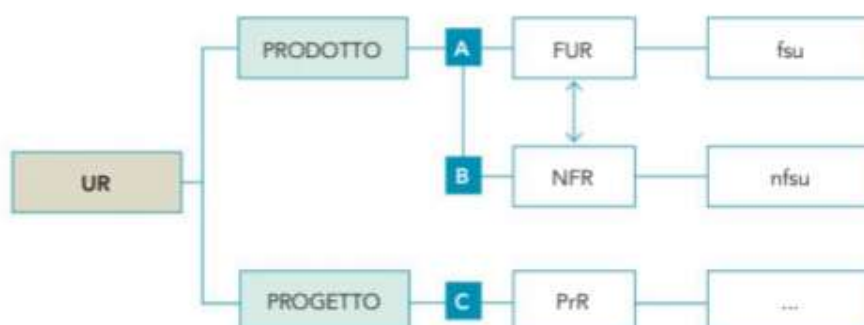


### ■ Attività correlate al progetto

Nelle fasi del ciclo di sviluppo del software, alcune attività sono **correlate alla misurazione funzionale** (analisi, implementazione, test di correttezza funzionale, ecc.), mentre altre sono **correlate alla verifica del suo funzionamento** (test di carico, di integrazione, collaudo, ecc.). Altre ancora sono **di natura progettuale** (coordinamento, formazione, ecc.). In **FIGURA 4** è riportato uno schema che prevede i **tre insiemi di attività**, elaborato dall'associazione GUFPI-ISMA, dedicata allo sviluppo di tecniche di misurazione del software.

**FIGURA 4** Insiemi di attività definite da GUFPI-ISMA. Nello schema i requisiti utente (UR, User Requirements), inerenti tanto al prodotto quanto al progetto, sono distinti in:

- **FUR** (Functional User Requirements), i requisiti relativi alle funzionalità che l'applicazione offre all'utente, misurati in **fsu** (Functional Size Unit), ovvero i function point;
- **NFR** (Non Functional Requirements), cioè le caratteristiche non funzionali, misurate in **nfsu** (Non Functional Size Unit);
- **PrR** (Project Requirements), attinenti alle attività di project management.



Nella **TABELLA 3** di seguito sono indicate le attività (discipline) secondo il paradigma RUP in un progetto di sviluppo del software (vedi Paragrafo 1.2). A ciascuna disciplina è associata una ipotesi di **percentuale di impegno sul progetto**. Le discipline prettamente funzionali, per le quali ha senso il calcolo in function point, pesano per l'80% del totale.

**TABELLA 3** Discipline secondo il paradigma RUP

|                       |                            |      |
|-----------------------|----------------------------|------|
| Discipline funzionali | Gestione requisiti         | ~20% |
|                       | Analisi e progettazione    | ~10% |
|                       | Implementazione            | ~35% |
|                       | Test funzionali            | ~15% |
| Discipline a supporto | Test non funzionali        | ~3%  |
|                       | Deployment & configuration | ~2%  |
|                       | Configuration management   | ~3%  |
|                       | Project management         | ~12% |
| Totale                |                            | 100% |

#### CASO

##### Effort (impegno) previsto

In riferimento all'applicazione OSMGB-Anagrafe Web vista in precedenza, è possibile stimare la previsione di impegno (effort) a partire dai function point (FP) totali e dalla **produttività** (cioè il numero di FP realizzati nell'unità di tempo) dell'azienda che realizza il progetto, a partire dai suoi dati storici.

Se i 310 FP stimati per il progetto corrispondono all'80% delle discipline complessive e la produttività dell'azienda è di 0,12 FP/ora, il numero di ore di impegno necessario per realizzare le discipline relative alla parte funzionale è dato dalla proporzione:

$$\text{produttività} : 1 \text{ ora} = \text{numero di FP} : x \text{ ore}$$

$$\text{quindi } 0,12 : 1 = 310 : x \quad \text{da cui} \quad x = \frac{310}{0,12} = 2.583$$

Per la realizzazione dell'80% del progetto si stima quindi un impegno complessivo di 2.583 ore. L'impegno complessivo del progetto (100%) sarà dunque dato dalla proporzione:

$$2.583 : 80 = x : 100 \quad \text{da cui} \quad x = 2.583 \times \frac{100}{80} = 3.229 \text{ ore}$$

Considerando un'attività lavorativa di otto ore al giorno, l'impegno risulta di 404 giorni, ovvero 81 settimane se si intende la settimana lavorativa di cinque giorni su sette. È importante non confondere l'impegno con il tempo complessivo stimato per concludere il progetto. Bisogna infatti considerare che il team di lavoro è composto da più persone che possono svolgere attività in contemporanea.

Sulla base di ciò, è possibile anche stimare gli impegni previsti per le singole discipline, come visibile nella tabella di seguito.

| Produttività funzionale/ore | FP          | Ore totali funzionali | Ore totali |
|-----------------------------|-------------|-----------------------|------------|
| 0,12                        | 310         | 2.583                 | 3.229      |
| Discipline                  |             | Ore                   | Giorni     |
| Gestione requisiti          | 20%         | 646                   | 81         |
| Analisi e progettazione     | 10%         | 323                   | 40         |
| Implementazione             | 35%         | 1.130                 | 141        |
| Test funzionali             | 15%         | 484                   | 61         |
| Test non funzionali         | 3%          | 97                    | 12         |
| Deployment & configuration  | 2%          | 65                    | 8          |
| Configuration management    | 3%          | 97                    | 12         |
| Project management          | 12%         | 388                   | 48         |
| <b>Totale</b>               | <b>100%</b> | <b>3.229</b>          | <b>404</b> |

## ■ Il costo del progetto

Se si possiede il dato sul costo orario di ogni singola figura professionale, grazie alla precedente misurazione è possibile effettuare una **stima del costo economico** complessivo del progetto.

Esistono delle **tabelle di riferimento** per identificare questi dati, per esempio nella Pubblica amministrazione, ma queste informazioni possono ovviamente cambiare nel tempo e in base al contesto di riferimento.

La misurazione effettuata rappresenta una stima non esente da margini di errore. Permette però di valutare in maniera oggettiva la dimensione dell'applicazione in termini di funzionalità offerte, indipendentemente dal contesto tecnologico di sviluppo.

### FISSA LE CONOSCENZE

1. Che cosa sono i SAL di progetto?
2. Che cosa sono i casi d'uso?
3. Quali sono gli ambienti operativi sui quali si installa il software?
4. Che cosa si intende per progettazione concettuale e logica di un database?
5. Come si può misurare il software?

## 5 CHIUSURA

### 5.1 Il rilascio

Tutti i progetti, a partire da quelli aziendali, nascono, evolvono e inevitabilmente si concludono. Nel presente caso il progetto deve essere testato e validato dal committente e dall'utente finale in una situazione di **rilascio** vera e propria: è necessario cioè effettuare una **installazione su un server** raggiungibile via web. Entro la data di conclusione deve inoltre anche essere svolto un lavoro di **presentazione** del progetto.

### 5.2 Il testing

#### ■ I test di sicurezza

Per garantire il buon funzionamento del sistema è necessario effettuare dei controlli (test) di varie tipologie a ogni singolo incremento di versione. Esistono strumenti che permettono lo svolgimento di **test automatici**, ma è altresì possibile svolgerli in **modalità manuale** se si hanno a disposizione una buona documentazione e un gruppo di lavoro apposito che se ne occupi costantemente.

- **Test delle funzionalità.** È utilizzato per verificare che ogni caso d'uso identificato nel documento di analisi sia stato provato e che svolga correttamente il proprio compito, sia nelle situazioni normali sia nei casi particolari.
- **Test di carico.** Ha il compito di verificare che il sistema sia in grado di funzionare anche con determinati carichi (numero di utenti, capacità del database, ecc.).
- **Test di usabilità o #accessibilità.** È usato per controllare se il sistema è in grado di garantire un buon grado di usabilità e se è accessibile.
- **Test di sicurezza.** È necessario per accertarsi che il sistema garantisca la sicurezza da attacchi informatici.

#### #techwords

##### ACCESSIBILITÀ

Proprietà che devono possedere i software per essere utilizzati con facilità dagli utenti, in particolare coloro che si trovano in condizioni di disabilità o di svantaggio.

##### FTP

Acronimo per File Transport Protocol. È un protocollo di comunicazione che permette il trasferimento di file.

### 5.3 L'installazione sul server

Una volta che si è verificato mediante i test che l'applicazione funziona correttamente nelle varie situazioni individuate dal documento dei piani di test (che è strettamente correlato al documento che descrive i casi d'uso), è possibile copiare il software sul server che ospita l'applicazione.

Normalmente per fare ciò si utilizza un client **#FTP** (un esempio è FileZilla) e si copiano i vari codici sorgente nelle cartelle opportune, esattamente come nell'ambiente di sviluppo.

### 5.4 La presentazione

Nella fase di chiusura del progetto è necessario prevedere una **presentazione conclusiva**, che potrà essere realizzata con PowerPoint, Prezi o altri strumenti di presentazione, anche online. In aggiunta o in alternativa, è possibile creare un breve **video** (circa 3 minuti) nel quale raccontare la storia del progetto, dall'idea alla sua realizzazione. Tutti i progetti correlati con OpenStreetMap vengono normalmente documentati sulle pagine wiki (<https://wiki.openstreetmap.org>).



È inoltre consigliabile aver predisposto un **sito web** (cosiddetto vetrina) per presentare il gruppo di lavoro e il progetto, nonché la documentazione; il sito vetrina ospiterà anche i video e le presentazioni iniziali e finali, e infine il link all'ultima versione del software. Il sito in questione potrà essere realizzato direttamente tramite codice sorgente HTML, ma anche grazie a uno dei tanti **#CMS** disponibili e gratuiti (per esempio Joomla! o Wordpress), oppure su piattaforme online condivise (come GitHub e Trello).

La redazione di una scheda di presentazione del progetto e un breve video sono richiesti anche nel caso di concorsi a premi (come [storiedialternanza.it](http://storiedialternanza.it), organizzato dalla Camera di Commercio), proposti da enti e associazioni in collaborazione con il MIUR. Aver realizzato una presentazione può essere inoltre un ottimo laboratorio per la preparazione all'esame di Stato, nel quale è ormai pratica consolidata dedicare uno spazio apposito alla discussione circa l'esperienza dei percorsi trasversali di orientamento (PCTO) e cittadinanza e costituzione.

Di seguito (FIGURA 5) è riportata una slide di esempio di una breve presentazione PowerPoint che è stata fatta all'interno della scuola.

Altri esempi di presentazioni sono disponibili ai seguenti link:

- [https://wiki.openstreetmap.org/wiki/IT:OsmGuineaBissau\\_Avogadro](https://wiki.openstreetmap.org/wiki/IT:OsmGuineaBissau_Avogadro);
- <https://www.youtube.com/watch?v=lj0iqdUjjAA>.

## #techwords

### CMS

Acronimo di Content Management System, si tratta di uno strumento software, installato su un server web, che facilita la gestione dei contenuti di siti web, anche a chi non ha conoscenze specifiche di programmazione.

FIGURA 5 Presentazione del progetto



## FISSA LE CONOSCENZE

1. Che cosa si intende per rilascio di un progetto?
2. Perché è importante effettuare i test di un sistema?
3. Quali tipologie di test vanno effettuate?
4. Quali sono le modalità per presentare un progetto?



## 5.5 Spunti per altri progetti

A questo punto è opportuno mettersi in gioco in prima persona con un progetto come quello presentato in questa Unità. Di seguito sono presenti alcuni spunti riguardanti il **volontariato**, l'**ambiente**, la **cultura** e il **territorio**. Ciò che conta, come accennato in precedenza, sono una buona idea e la motivazione nell'intraprendere il progetto.

### ■ Volontariato

Cercate un'associazione o un ente di volontariato vicino a voi. Probabilmente avranno un sito web da migliorare o la richiesta di un sistema informativo efficiente per raccogliere e organizzare i propri dati. Potrebbero chiedervi di gestire o mappare i loro progetti umanitari, le risorse materiali o altro ancora.

- **Scheda sanitaria web.** Un'idea per un progetto potrebbe essere la realizzazione di un'applicazione per la gestione di schede sanitarie per tener traccia dei disturbi, delle malattie e delle vaccinazioni dei loro assistiti. Molte associazioni che operano in zone in cui non esistono ancora veri e propri sistemi di raccolta dati potrebbero collaborare nelle fasi di analisi e svolgere il ruolo di committenti.
- **Mappa degli interventi.** Un progetto potrebbe prevedere di mappare su OpenStreetMap tutti gli interventi umanitari di un'associazione in Italia, in Europa o in aree remote del mondo. È possibile farlo usando Umap (<https://umap.openstreetmap.fr/it/>), che permette di creare mappe personalizzate.

### ■ Ambiente

È importante poter conoscere la qualità dell'aria che viene respirata tutti i giorni. Un'idea per un progetto potrebbe essere quella di effettuare l'**analisi della temperatura o delle sostanze presenti nell'aria all'esterno o all'interno della scuola** grazie a sistemi di rilevazione (sensori) collegabili per esempio con PC Arduino o Raspberry. I dati così rilevati possono poi essere archiviati su un database ed essere resi disponibili su un sito web tramite grafici, statistiche e riferimenti specifici alle normative vigenti riguardo la sicurezza e la salute.

### ■ Cultura e territorio

Perché non prendersi cura di un **museo**, un **monumento** o un **edificio storico**, magari nelle vicinanze della scuola, e avviare una ricerca di **monitoraggio civico**, volta per esempio all'**analisi dei fondi spesi e degli investimenti economici fatti nel corso degli anni**?

Con questa finalità è nato un progetto, chiamato "a scuola di opencoesione", organizzato da ASOC, un ente istituzionale che collabora con varie istituzioni.

Il progetto permette di sviluppare competenze digitali, statistiche e di educazione civica per aiutare gli studenti a conoscere e comunicare, con l'ausilio di tecniche giornalistiche, in che modo le politiche pubbliche, e in particolare le politiche di coesione, intervengano nei luoghi dove vivono.

Si può così conoscere il mondo degli **#open data** e del **#data journalism** e imparare a utilizzare tecniche di comunicazione e strumenti di infografica.

Per informazioni, visitate il sito: [www.ascuoladiopencoesione.it](http://www.ascuoladiopencoesione.it)

#### #techwords

##### OPEN DATA

Sono dati liberamente accessibili che possono essere riutilizzati, a patto che siano mantenuti aperti ed eventualmente ne sia citata la fonte.

##### DATA JOURNALISM

Si tratta della pratica di giornalismo in cui vengono impiegati strumenti della matematica, della statistica e delle scienze sociali e comportamentali.

## 1 La sperimentazione in classe di un caso concreto

Un caso concreto di realizzazione di un progetto software è rappresentato dall'esperienza di una classe quinta specializzazione informatica che ha collaborato con un'associazione di volontariato, operante in Guinea Bissau, per mappare su OpenStreetMap (OSM) abitazioni e punti di interesse di uno specifico villaggio e per realizzare un'applicazione web per la gestione della sua anagrafica.



## 2 Avvio

Il progetto nasce da un'idea e da un bisogno. Stipulato il contratto tra le parti interessate (stakeholder), inizia con un kick-off e con la redazione del project chart. Stabilita la fattibilità, si deve conferirgli un'identità, per presentarlo al committente tramite un elevator pitch.

## 3 Approfondimento

In questa fase, viene definito il contesto del progetto. Nel corso della realizzazione, vengono redatti dei verbali. A questo punto si definiscono la WBS (Work Breakdown Structure) e il diagramma di Gantt. Viene organizzata la gestione della documentazione di progetto e il gruppo di lavoro con le rispettive responsabilità.

## 4 Realizzazione

In questa fase, con gli Stati Avanzamento Lavori (SAL), vengono dettagliati l'analisi dei requisiti, i casi d'uso, l'architettura del sistema software ed eventualmente un prototipo di interfaccia utente. Viene inoltre definita la gestione delle versioni. Vengono progettati il database (a livello concettuale, logico e poi fisico) e le prime versioni del software. Il progetto viene monitorato dal punto di vista di tempi e costi e da quello tecnico tramite test e metriche funzionali come i function point.



## 5 Chiusura

Per presentare il progetto all'esame di Stato, è necessario testarlo e installarlo su un vero ambiente di produzione. Gli studenti devono infine saperlo presentare tramite un elevator pitch, delle slide o un sito web vetrina.

