



# APPLICAZIONE DI ALLERTA PER LA PROTEZIONE CIVILE



Politecnico di Milano

• Ingegneria Industriale e dell'informazione •

Tesi di Laurea

Anno accademico 2019/2020

Di Falco  
Maria Chiara

Bonanomi  
Mirko

Bancalè  
Lorenzo

# Indice

<b>1. Testo del progetto</b>	
1.1 Tema e specifiche.....	3
<b>2. Studio di fattibilità</b>	
2.1 Analisi di costo, benefici e rischio.....	5
2.2 Commento al Goal Diagram	
2.2.1 Sistema centrale.....	7
2.2.2 Applicazione di previsione.....	8
2.2.3 Utente.....	9
2.3 Data dictionary.....	10
<b>3. Design</b>	
3.1 Class Diagram.....	12
3.2 Object Diagram.....	15
3.3 Use Case Diagram.....	16
3.4 Sequence Diagram : generazione e registrazione previsione.....	17
3.5 Activity Diagram : generazione e registrazione previsione.....	18
3.6 State Diagram : ricezione salvataggio previsione.....	19
3.7 Sequence Diagram : ricerca completa di previsioni, con tutti i possibili filtri.....	20
3.8 Activity Diagram : invio notifiche.....	22
3.9 State Diagram : invio notifiche.....	22
3.10 Deployment Diagram.....	23
3.11 Component Diagram.....	24
3.12 Collaboration Diagram.....	25
<b>4. Implementazione</b>	
4.1 <b>Commento al codice Java</b>	
4.1.1 Packages e classi.....	27
4.1.2 Test JUnit.....	28
4.2 <sup>2</sup> Commento all'applicazione.....	30

# Testo del progetto

## Tema e Specifiche

Attraverso un sistema di allarmi definiti secondo una scala di gravità (1-10), la Protezione Civile dà l'allerta per fenomeni di maltempo (piogge violente, nevicate intense, forte vento o grandine) e terremoti.

Le allerte sono riferite ai codici di avviamento postale (CAP), ad ognuno dei quali è correlata una lista di utenti. I dati sono mantenuti costantemente aggiornati dal Sistema Centrale sulla base degli eventi (previsioni e accadimenti) ricevuti dalle applicazioni di previsione distinte in due centri specializzati: 3BMeteo per le previsioni meteo e INGV (Istituto Nazionale di Geofisica e Vulcanologia) per le previsioni relative a terremoti; questi elaborano ogni 30 minuti (implementati come 2,5 secondi nel codice) le previsioni per i diversi fenomeni e comunicano gli accadimenti ogni ora (implementati come 5 secondi nel codice).

Il sistema registra tutte le previsioni dei fenomeni/eventi. Ogni evento è descritto da un ID, un CAP, una sorgente, un tipo, la data e l'ora; inoltre per le previsioni viene definito anche un livello di gravità.

Ogni 4 ore (implementate come 1 minuto nel codice) vengono prelevate le previsioni presenti per ogni CAP e vengono riordinate per gravità; ciò permette di definire un elenco di allarmi in evidenza, che sono quelli di massima gravità (livello di gravità  $\geq 6$ ) per fascia oraria e per CAP. Tale elenco viene inoltrato a tutti gli utenti relativi ad un certo CAP. Se in base a previsioni successive un'allerta rientra, questa viene rimossa dalle allerte in evidenza.

Gli utenti possono cercare gli eventi per istante temporale, per tipo e/o per CAP, ottenuti dalla lista di eventi correnti, ovvero quelli in corso per ogni CAP nel giorno corrente, o tra quelli presenti nello storico (accadimenti).

# STUDIO DI FATTIBILITA'

## Introduzione

Lo studio di fattibilità di questo progetto è strutturato nel modo seguente: si analizza il contesto nel quale il progetto è situato, delineandone approssimativamente la strutturazione e gli obiettivi, si valutano i rischi e quindi il rapporto costi-benefici, e, infine, si forniscono eventuali raccomandazioni per le fasi realizzative.

## Contesto dello studio

Il progetto che si sta analizzando riguarda la realizzazione di un sistema software per la Protezione Civile in grado di notificare una previsione meteorologica o di altri tipi di rischi per i cittadini di un determinato comune. Attualmente la Protezione Civile non dispone di un tale applicativo software in grado di avvisare tempestivamente l'utente dell'arrivo di un fenomeno, ad esempio di maltempo, di una certa rilevanza.

## Definizione degli obiettivi del progetto

Lo scopo di tale progetto è quello di allertare i cittadini nel momento in cui vengono rilevati eventi per essi rischiosi segnalando il tipo di fenomeno, l'orario di previsione di questo e altri fattori. Gli utenti possono inoltre rimanere costantemente aggiornati sugli eventi in corso, previsti o passati, consultando un elenco di questi; tale elenco è gestito e organizzato da un sistema centrale che comunica direttamente con le applicazioni di previsione e di rilevazione fenomeni.

## Analisi dei benefici

Il software offre un servizio aggiuntivo rispetto a quelli standard forniti dalla Protezione Civile; questa ha infatti lo scopo generale di salvaguardare la sicurezza dei cittadini e il software in analisi permettere di raggiungere tale scopo con minore difficoltà e maggiore efficienza.

In caso di uragani, ad esempio, un'allerta tempestiva e sommariamente precisa aumenta di gran lunga il tempo disponibile per l'evacuazione e ciò permette soprattutto di limitare eventuali danni al territorio e di diminuire il numero di feriti.

## Analisi del rischio

L'importanza di questa analisi deriva dal fatto che la mancata individuazione dei rischi del progetto, potrebbe comportare l'insuccesso di questo.

Uno dei possibili rischi di tale progetto consiste nella non accuratezza delle previsioni: più alta è la percentuale di previsioni errate, maggiori saranno gli allarmi forniti inutilmente causando disagio ai cittadini. Questo implica un alto tasso di inconsistenza delle informazioni e di conseguenza una diminuzione della credibilità e quindi dell'utilizzo del software.

Tuttavia tale rischio non è così alto da definire infattibile la progettazione del software in questione. Esistono poi altri rischi che si riscontrano in qualsiasi progettazione software (tra cui, ad esempio, la mancanza di esperienza sui sistemi correlati al progetto da parte dei programmatori) che pertanto non vengono considerati di totale impedimento alla realizzazione del progetto.

## Costi del progetto

Per l'analisi dei costi è stata stilata una tabella riepilogativa che definisce le principali voci di costo del progetto, relative alla realizzazione e alla messa in opera del nuovo sistema.

<b>COSTI DI PROGETTO</b>	
<b>Costi di costruzioni (esterni)</b>	<b>Software:</b> - Sviluppo Database condiviso 1500€ - Licenza Sql : 930 € - Licenze Microsoft Windows 10 (100€ cad) <b>Hardware:</b> - Desktop pc: 1000 € - Sensori di rilevazione dati meteorologici (dell'ordine delle migliaia di euro)
<b>Costi di costruzioni (esterni)</b>	- Revisione e valutazione delle strumentazioni disponibili: 35€/ora
<b>Costi di avviamento (interni)</b>	- Configurazione applicativo presso Protezione Civile: 35 € / ora - Costo formazione personale che utilizzerà l'applicazione: 35 € / ora
<b>COSTI DI PROGRAMMAZIONE E MANUTENZIONE</b>	
<b>Costi di programmazione applicativo</b>	- Programmatore: 30 € / ora - Testing unitario: 30 € / ora
<b>Test</b>	- Testing di Sistema (sul campo): 35 € / ora
<b>Manutenzione</b>	- Manutenzione ordinaria software applicativo: 35€ / ora - Manutenzione sistematica ordinaria: 35€ / ora

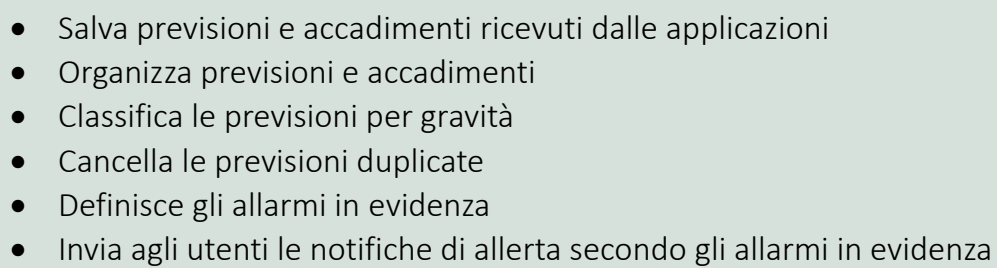
A questi possono essere aggiunti costi ulteriori (dell'ordine delle centinaia di euro) per l'utilizzo di strumentazioni più accurate che permettano di ridurre il rischio sopra spiegato di inconsistenza delle previsioni

## Conclusione

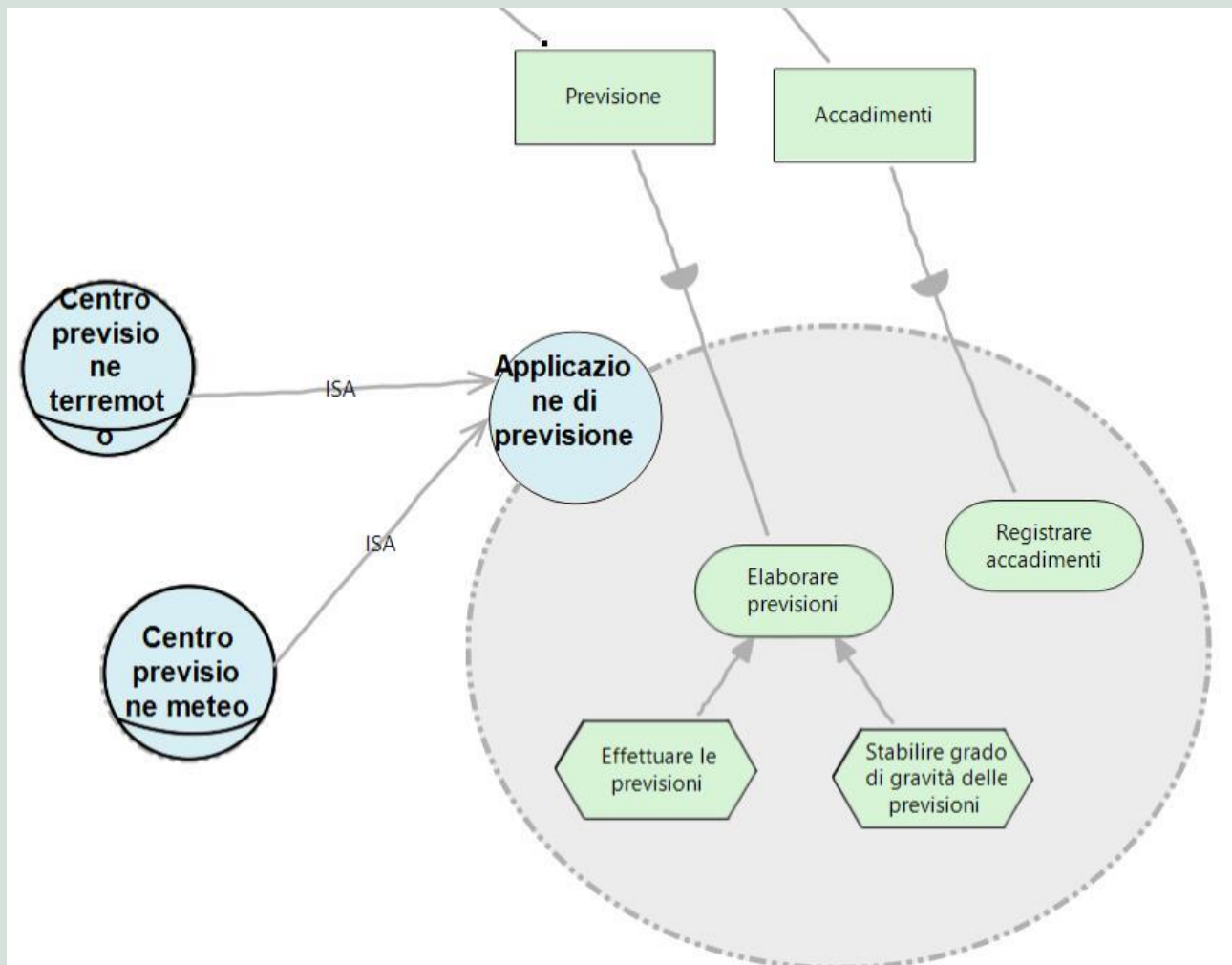
Con questo studio si crede quindi di aver previsto i possibili scenari che si potrebbero presentare nel corso della progettazione e che potrebbero comprometterne la realizzazione.

Si conclude quindi dichiarando questo progetto fattibile.

## Sistema centrale



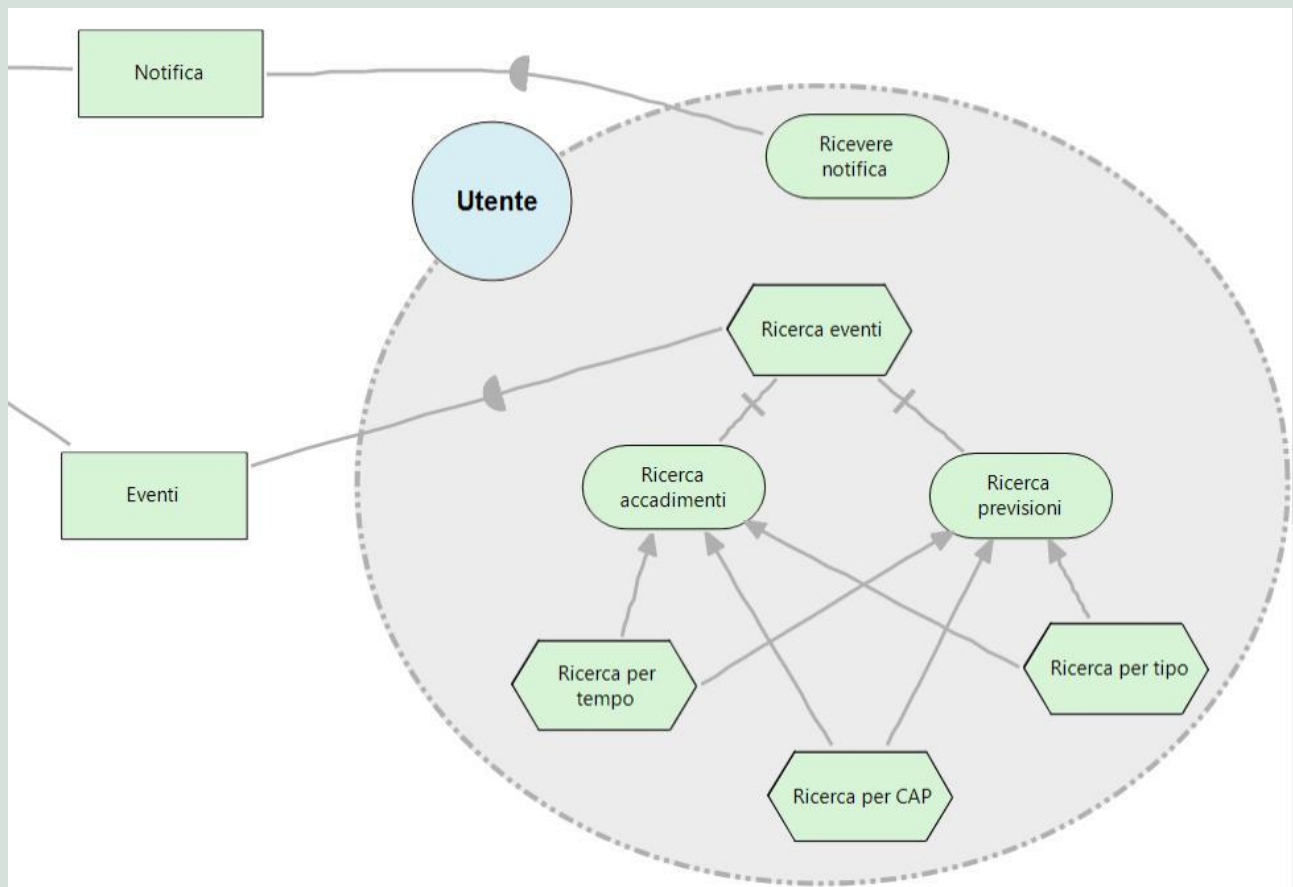
## Applicazione di previsione



- Può essere un Centro di previsione terremoti o Centro di previsione meteo
- Elabora le previsioni effettuate associandogli un grado di gravità
- Invia le previsioni elaborate al Sistema Centrale
- Registra gli accadimenti
- Invia gli accadimenti al Sistema Centrale



# Utente



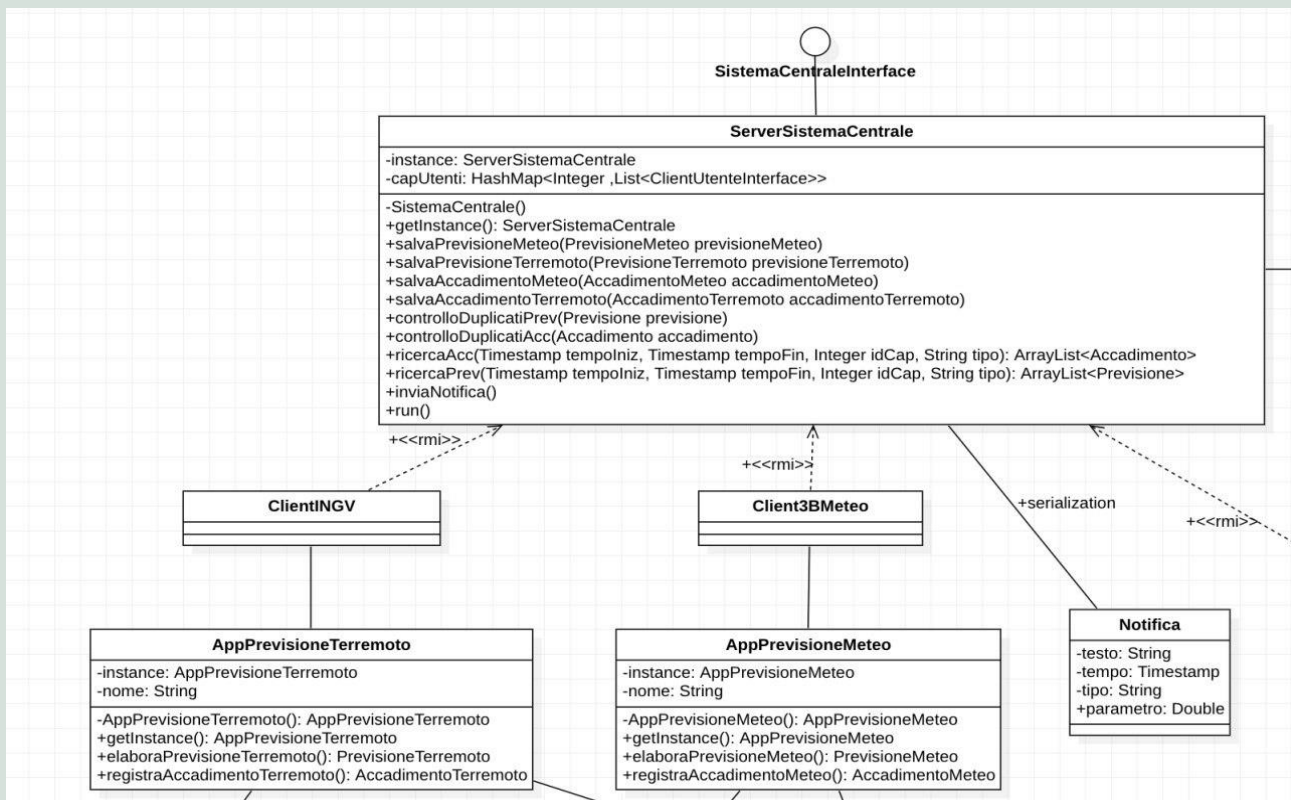
- Riceve le notifiche di previsioni (allerte) dal Sistema Centrale
- Ricerca gli eventi (previsioni o accadimenti) per tempo, CAP o tipo.

# DATA DICTIONARY

NOME	DESCRIZIONE	SINONIMI	ESEMPI	SOTTOTIPI	SUPERTIPI	ATTRIBUTI	COMPONENTI	RELAZIONI
<b>Evento</b>	Termine generico che può indicare una previsione o un accadimento	-Fenomeno	-E' prevista pioggia presso il cap 22030 in data 21 maggio alle ore 16:15 con precipitazione media di 30mm/h --Si è verificata un terremoto presso il cap 23895 in data 21 maggio alle ore 18.00 con magnitudo 1,7 della scala Richter	-Previsione -Accadimento		-IdEvento -CAP -Sorgente -Tipo -Tempo		
<b>Previsione</b>	Supposizione relativa ad eventi futuri	-Predizione	-E' prevista pioggia presso il cap 22030 in data 21 maggio alle ore 16:15 con precipitazione media di 30mm/h	-Previsione meteo -Previsione terremoto	-Evento	-LivelloGravità		
<b>Previsione meteo</b>	Previsione meteorologica	-Predizione meteorologica	-E' prevista pioggia presso il cap 22030 in data 21 maggio alle ore 16:15 con precipitazione media di 30mm/h		-Previsione	-Precipitazione		
<b>Previsione terremoto</b>	Previsione terremoto	-Predizione terremoto	-E' previsto terremoto presso il cap 23895 in data 21 maggio alle ore 16:15 con precipitazione media di 2,0 della scala Richter		-Previsione	-Magnitudo		
<b>Accadimento</b>	Evento effettivamente accaduto	-Avvenimento	--Si è verificato un terremoto presso il cap 23895 in data 21 maggio alle ore 18.00 con precipitazione media di 30mm/h	-Accadimento meteo -Accadimento terremoto	-Evento			
<b>Accadimento meteo</b>	Accadimento meteorologico	-Avvenimento meteorologico	--Si è verificata neve presso il cap 23895 in data 21 maggio alle ore 18.00 con precipitazione media di 0,2cm/h		- Accadimento	-Precipitazione		
<b>Accadimento terremoto</b>	Accadimento terremoto	-Avvenimento terremoto	--Si è verificato un terremoto presso il cap 23895 in data 21 maggio alle ore 18.00 con magnitudo 3,2 della scala Richter		- Accadimento	-Magnitudo		
<b>Notifica</b>	Avviso di una previsione di alta gravità	-Allarme	-ATTENZIONE: Si prevede vento in data 23 maggio 2020 alle ore 8:00 con velocità media di 130km/h			-Testo -Tempo -Tipo -Parametro		-Utente
<b>Utente</b>	Persona che utilizza l'applicazione	-Abitante	-Marioi, 23895			-Nome -CAP		-Notifica

# DESIGN

# Class Diagram



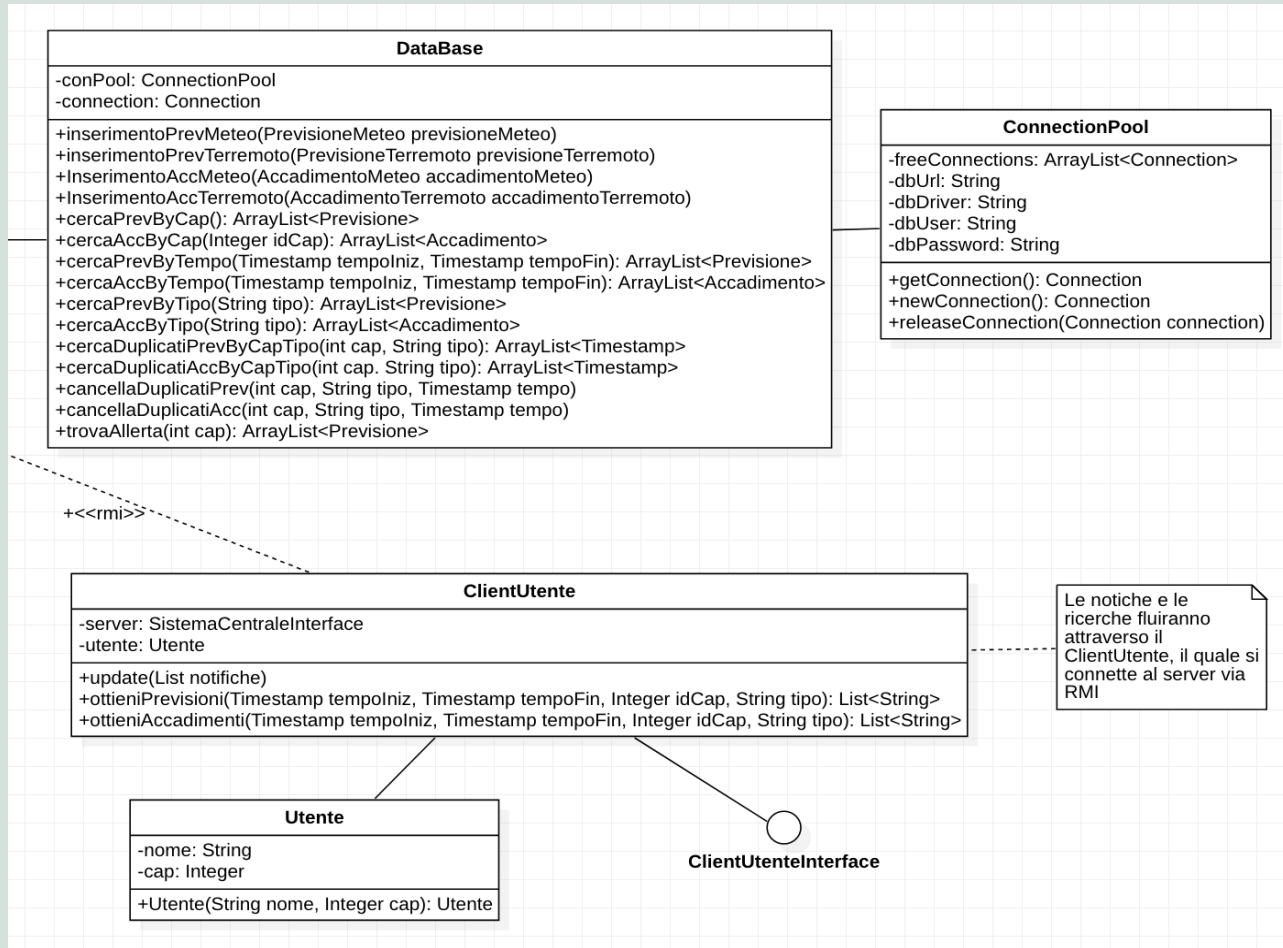
Nel class diagram possiamo vedere le classi con relativi attributi, operazioni e relazioni.

Le classi sono:

- **AppPrevisioneTerremoto**: è la classe che rappresenta l'applicazione che fornisce le previsioni (**elaboraPrevisioneTerremoto**) e gli accadimenti (**registraAccadimentoTerremoto**) di tipo terremoto; per farlo, chiama i costruttori delle relative classi "PrevisioneTerremoto" e "AccadimentoTerremoto"; in particolare si occupa di assegnare un livello di gravità alle previsioni in base al loro valore di magnitudo.
- **AppPrevisioneMeteo**: è la classe che rappresenta l'applicazione che fornisce le previsioni (**elaboraPrevisioneMeteo**) e gli accadimenti (**registraAccadimentoMeteo**) di tipo meteo; per farlo, chiama i costruttori delle relative classi "PrevisioneMeteo" e "AccadimentoMeteo" assegnando poi successivamente un valore di precipitazione/velocità a seconda del tipo di evento (pioggia, neve, vento o grandine); in particolare si occupa di assegnare un livello di gravità alle previsioni in base al valore di precipitazione/velocità.
- **ServerSistemaCentrale**: è la classe che rappresenta il sistema che raccoglie tutte le previsioni (**salvaPrevisioneMeteo** e **salvaPrevisioneTerremoto**) e tutti gli accadimenti (**salvaAccadimentoMeteo** e **salvaAccadimentoTerremoto**) che riceve dalle applicazioni di previsione; si occupa anche di cancellare, ad ogni inserimento di una nuova previsione o di un nuovo accadimento, previsioni (**controllaDuplicatiPrev**) o accadimenti (**controllaDuplicatiAcc**) dello stesso tipo, appartenenti allo stesso cap e relativi alla stessa fascia oraria (sarebbe incoerente, ad esempio, avere due previsioni per un certo cap, di un certo tipo e relative ad una certa fascia oraria, che abbiano due livelli di gravità completamente diversi; di conseguenza si mantiene la previsione più recente poiché considerata la più attendibile). Inoltre, contiene i metodi per registrare nuovi utenti

(registraUtente) e per elaborare per ogni cap le notifiche (inviaNotifica) da inviare agli utenti (salvati nella hashmap "capUtenti"). Infine, contiene i metodi per la ricerca di previsioni (ricercaPrev) e accadimenti (ricercaAcc).

- Notifica: è la classe che definisce la struttura delle notifiche; ogni notifica è caratterizzata da un testo e dai parametri che descrivono l'allerta.

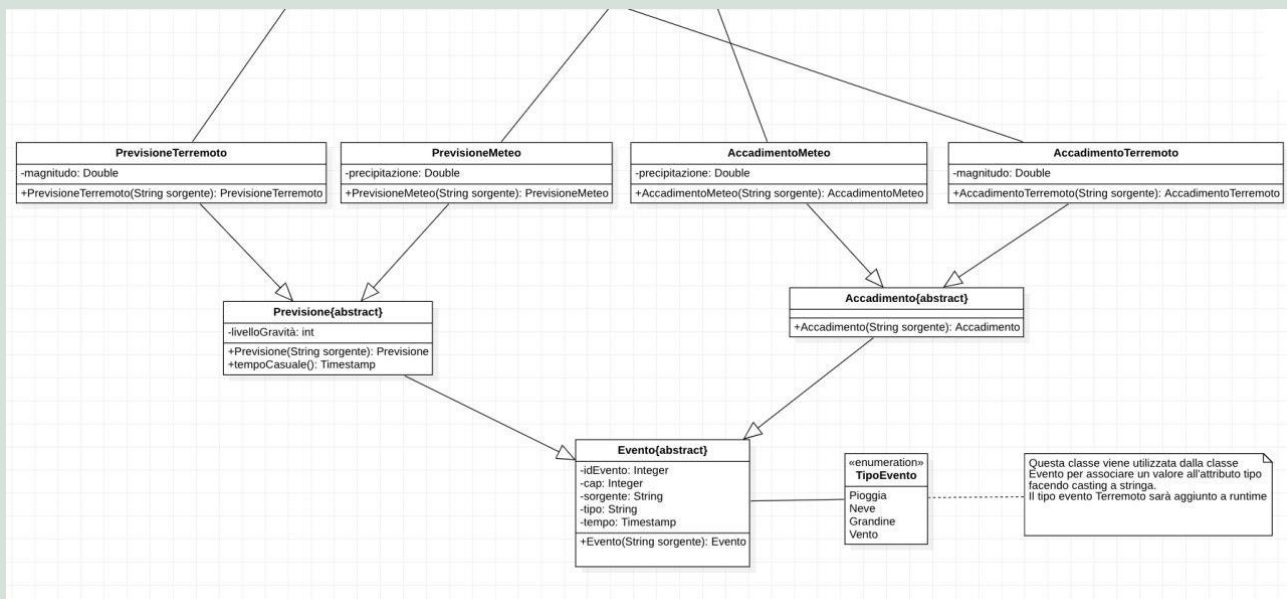


- DataBase: è la classe che contiene i metodi che si occupano di effettuare le query al database per inserire previsioni o accadimenti (inserimentoPrevMeteo, inserimentoPrevTerremoto, inserimentoAccMeteo, inserimentoAccTerremoto), trovare e cancellare eventuali duplicati (cercaDuplicatiPrevByCapTipo, cercaDuplicatiAccByCapTipo, cancellaDuplicatiPrev e cancellaDuplicatiAcc), ricercare previsioni e accadimenti secondo determinati filtri (cercaPrevByCap, cercaAccByCap, cercaPrevByTempo, cercaAccByTempo, cercaPrevByTipo e cercaAccByTipo) e trovare le previsioni da comunicare all'utente come allerte (trovaAllerta).

- ConnectionPool: è la classe che si occupa di gestire le connessioni al database (getConnection, newConnection e releaseConnection).

- ClientUtente: è la classe che si occupa di mostrare a video (sull'interfaccia grafica) le notifiche ricevute dal sistema centrale (update); si occupa inoltre di chiamare i metodi di ricerca del sistema centrale quando viene cliccato il tasto di ricerca nell'interfaccia grafica (ottieniPrevisioni e ottieniAccadimenti).

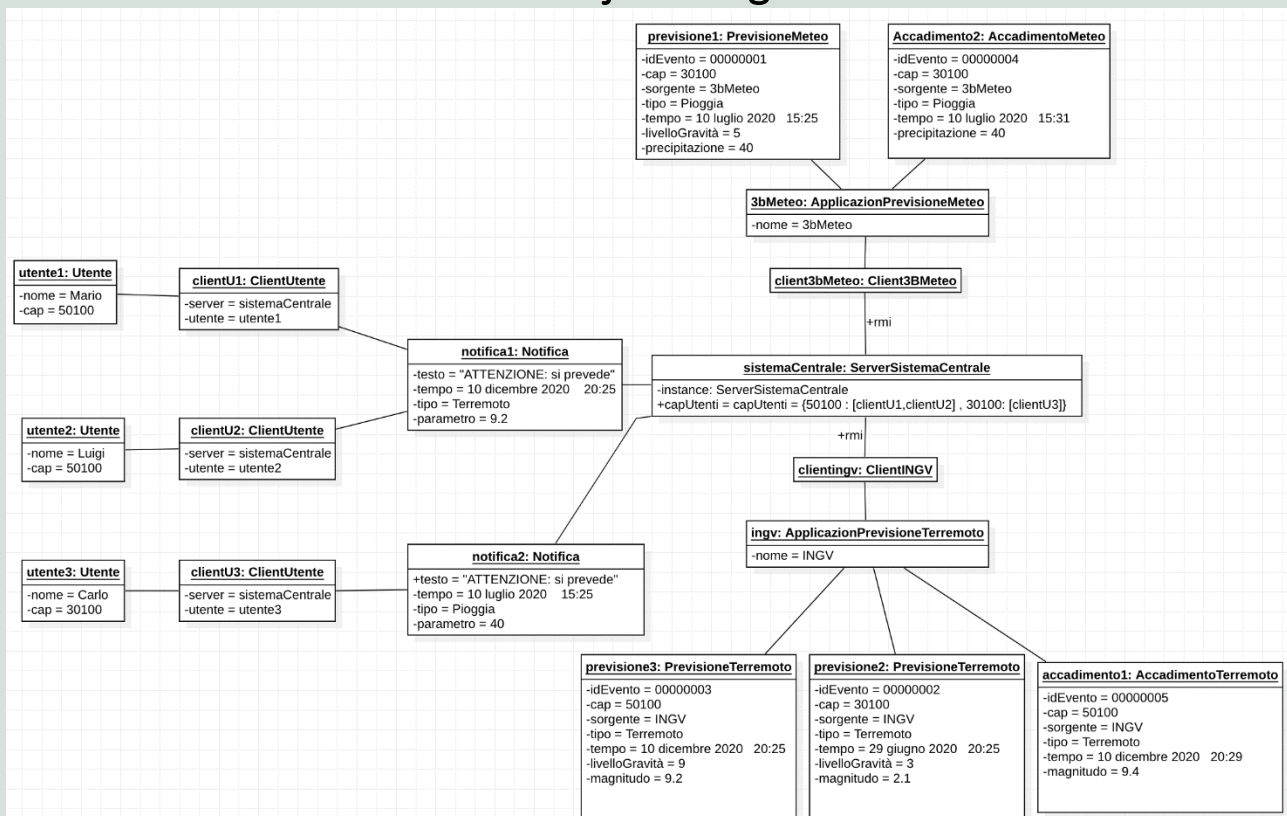
- Utente: è la classe che definisce gli attributi di un utente.



- Evento: è la classe che definisce le caratteristiche di un evento ed è quindi la superclasse di tutte le classi relative agli eventi;
- Previsione: è la classe che definisce le caratteristiche di una previsione, prelevando al contempo quelle di evento, di cui è sottoclasse.
- Accadimento: è la classe che definisce le caratteristiche di un accadimento, prelevando al contempo quelle di evento, di cui è sottoclasse.
- PrevisioneMeteo: è la classe che definisce le caratteristiche di una previsione di tipo meteo, prelevando al contempo quelle di previsione, di cui è sottoclasse.
- PrevisioneTerremoto: è la classe che definisce le caratteristiche di una previsione di tipo terremoto, prelevando al contempo quelle di previsione, di cui è sottoclasse.
- AccadimentoMeteo: è la classe che definisce le caratteristiche di un accadimento di tipo meteo, prelevando al contempo quelle di accadimento, di cui è sottoclasse.
- AccadimentoTerremoto: è la classe che definisce le caratteristiche di un accadimento di tipo terremoto, prelevando al contempo quelle di accadimento, di cui è sottoclasse.



# Object Diagram

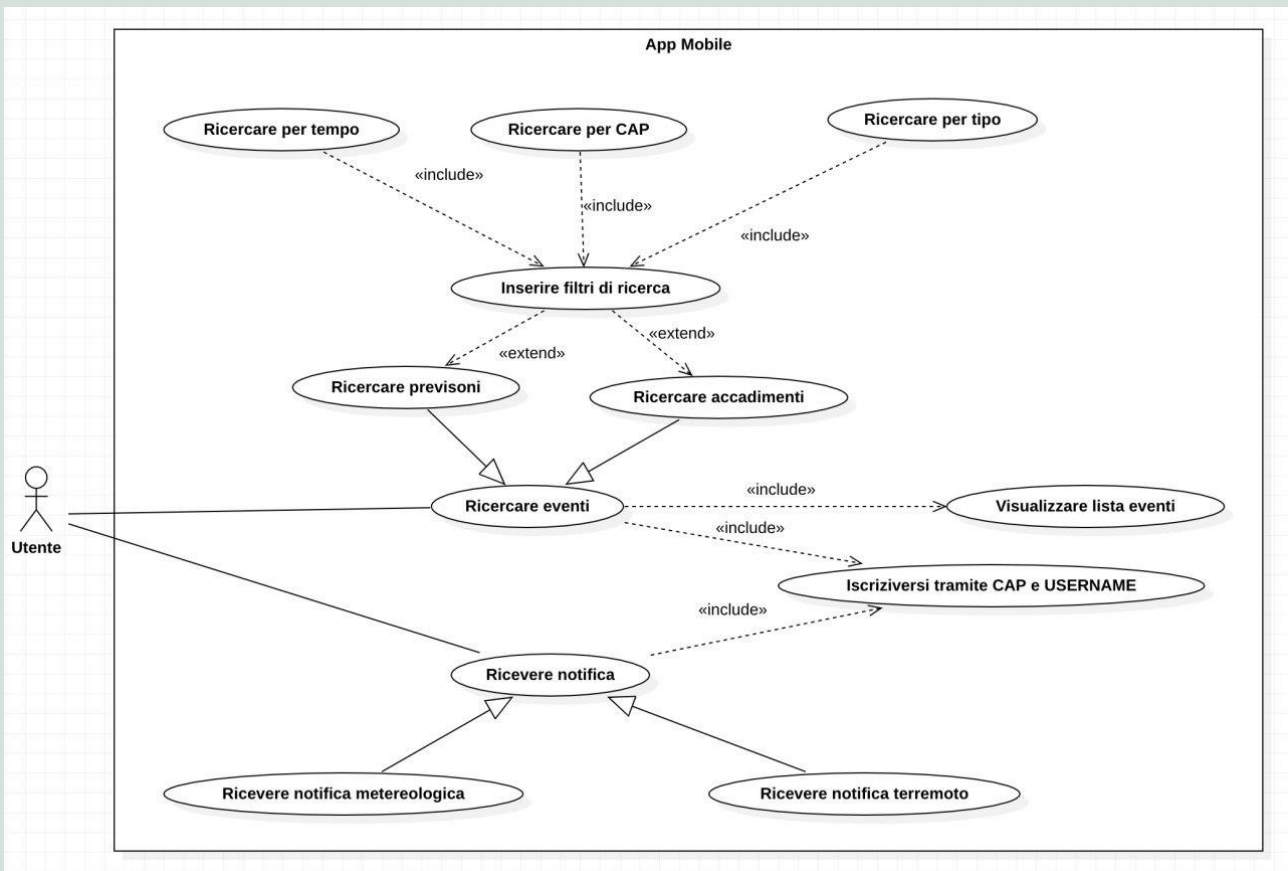


L'object diagram rappresenta un insieme di possibili istanze delle classi.

Nell'esempio qui riportato sono presenti:

- l'istanza (unica) della classe SistemaCentrale;
- l'istanza (unica) della classe ApplicazionePrevisioneMeteo;
- l'istanza (unica) della classe ApplicazionePrevisioneTerremoto;
- le istanze (uniche) dei client relativi alle applicazioni di previsione che comunicano col server;
- 3 istanze della classe Utente, ognuna col proprio nome, di cui 2 relativi al cap 50100 e uno relativo al cap 30100;
- 3 istanze della classe ClientUtente, ognuna relativa a una delle istanze della classe Utente;
- 2 istanze della classe PrevisioneTerremoto, che rappresentano appunto due previsioni di tipo terremoto, di cui una relativa al cap 50100 di gravità 9 e una relativa al cap 30100 di gravità 3;
- una istanza della classe AccadimentoTerremoto, che rappresenta appunto un accadimento di tipo terremoto relativo al cap 50100;
- una istanza della classe PrevisioneMeteo che rappresenta una previsione di tipo pioggia relativa al cap 30100;
- una istanza della classe AccadimentoMeteo che rappresenta un accadimento di tipo pioggia relativo al cap 30100;
- 2 istanze della classe Notifica relative alle due allerte da comunicare al cap 50100 e al cap 30100, ovvero rispettivamente la previsione di terremoto di magnitudo 9.2 e la previsione di pioggia di 40 mm/h (sono le due previsioni di massima gravità).

# Use Case Diagram

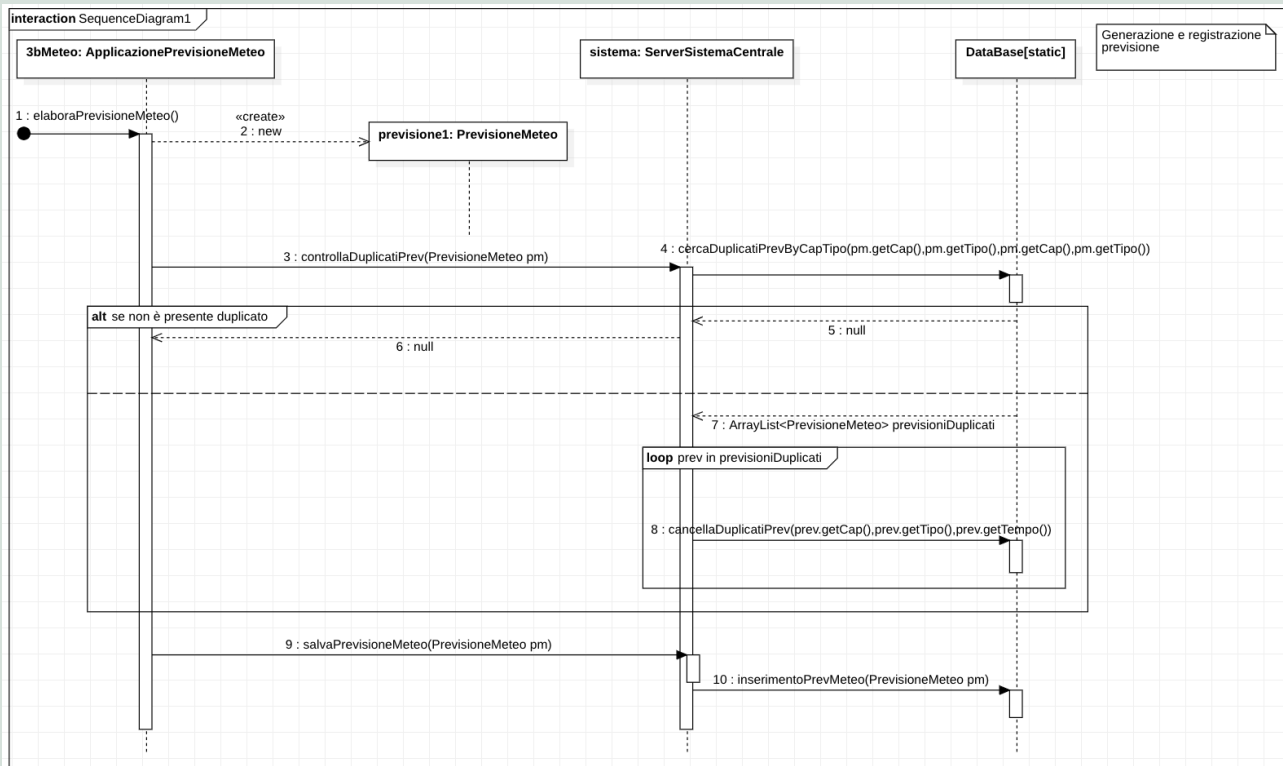


Lo use case diagram permette di definire i tipici casi d'uso del sistema, ovvero un servizio che questo offre a uno o più attori.

Nel nostro caso gli attori sono gli utenti, i quali, tramite l'applicazione, ricevono le notifiche (allerte) relative a previsioni di massima gravità e possono ricercare previsioni o accadimenti secondo determinati filtri (tempo, tipo e/o cap). Per poter compiere entrambe le operazioni, è necessario che l'utente sia iscritto all'applicazione tramite un cap e uno username



## Sequence Diagram: generazione e registrazione previsione



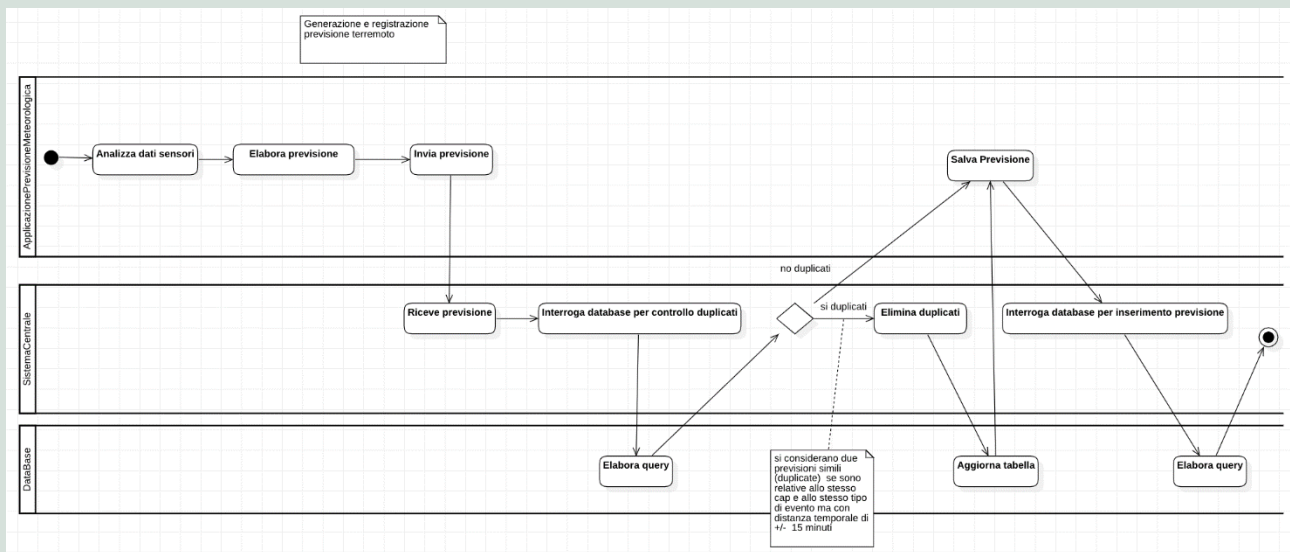
Il sequence diagram permette di mostrare i passaggi sequenziali compiuti dal programma per svolgere un determinato processo, mettendo in evidenza i metodi che servono e un possibile insieme di istanze delle classi a cui questi appartengono.

Nel caso in questione, il processo descritto è la generazione di una previsione di tipo meteo e il suo salvataggio nel database.

L'applicazione meteo ha il compito di generare una nuova previsione meteo chiamando il costruttore della classe omonima; poi chiama il metodo del server che controlla la presenza di eventuali duplicati della previsione appena generata, il quale a sua volta chiama un metodo della classe database che permette di prelevare dal database le previsioni che potrebbero risultare essere dei duplicati. A questo punto, se sono presenti dei duplicati, il server si occupa di eliminarli chiamando una funzione apposita dal database, altrimenti rimanda il controllo all'applicazione meteo.

Infine, l'applicazione chiama la funzione del server che permette di salvare la previsione, il quale a sua volta chiama la funzione della classe database che inserisce la nuova previsione nel database.

## Activity Diagram: generazione e registrazione previsione terremoto



L'activity diagram permette di descrivere un processo attraverso dei grafi costituiti da nodi, che definiscono le attività, e da archi che stabiliscono l'ordine di svolgimento delle attività. Inoltre, è possibile definire le classi responsabili delle varie attività tramite rettangoli, ognuno dei quali si riferisce ad una specifica classe e contiene appunto le attività ad essa delegate.

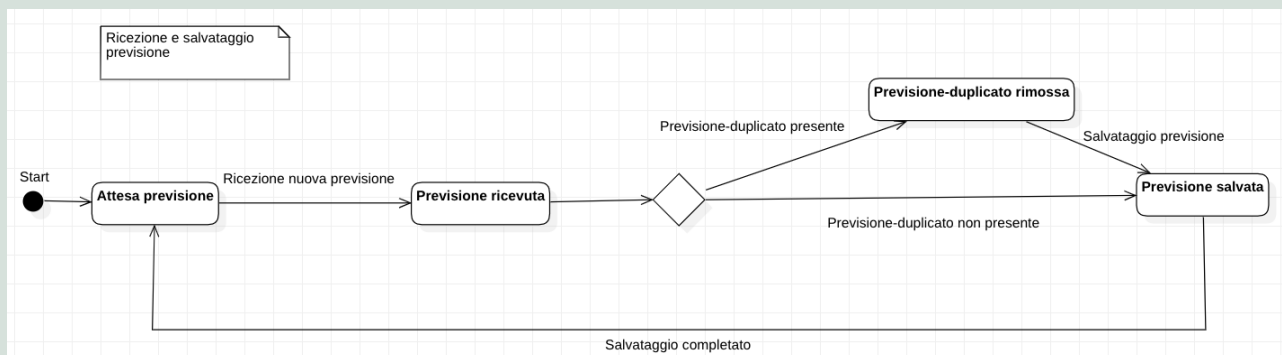
Nel caso in questione, il processo descritto riguarda la generazione di una previsione di tipo terremoto e il relativo salvataggio.

La prima attività viene svolta dall'applicazione di previsione terremoto e riguarda l'analisi dei dati che questa preleva da (ipotetici) sensori; dai dati estrapolati, l'applicazione elabora la previsione e la invia al sistema centrale. Questo riceve quindi la previsione e si rivolge alla classe database per valutare l'eventuale presenza di previsioni che risultino essere duplicate di quella ricevuta; tale classe interroga quindi il database elaborando la query necessaria e ritorna il risultato al sistema centrale. Se il risultato è vuoto, ovvero non è stato trovato nessun duplicato, allora il sistema centrale passa direttamente il controllo all'applicazione di previsione, altrimenti si occupa prima di cancellare i duplicati.

Infine, l'applicazione di previsione richiede al sistema centrale il salvataggio della nuova previsione.

Quindi il processo termina aggiornando la tabella delle previsioni nel database.

## State Diagram: ricezione e salvataggio previsione

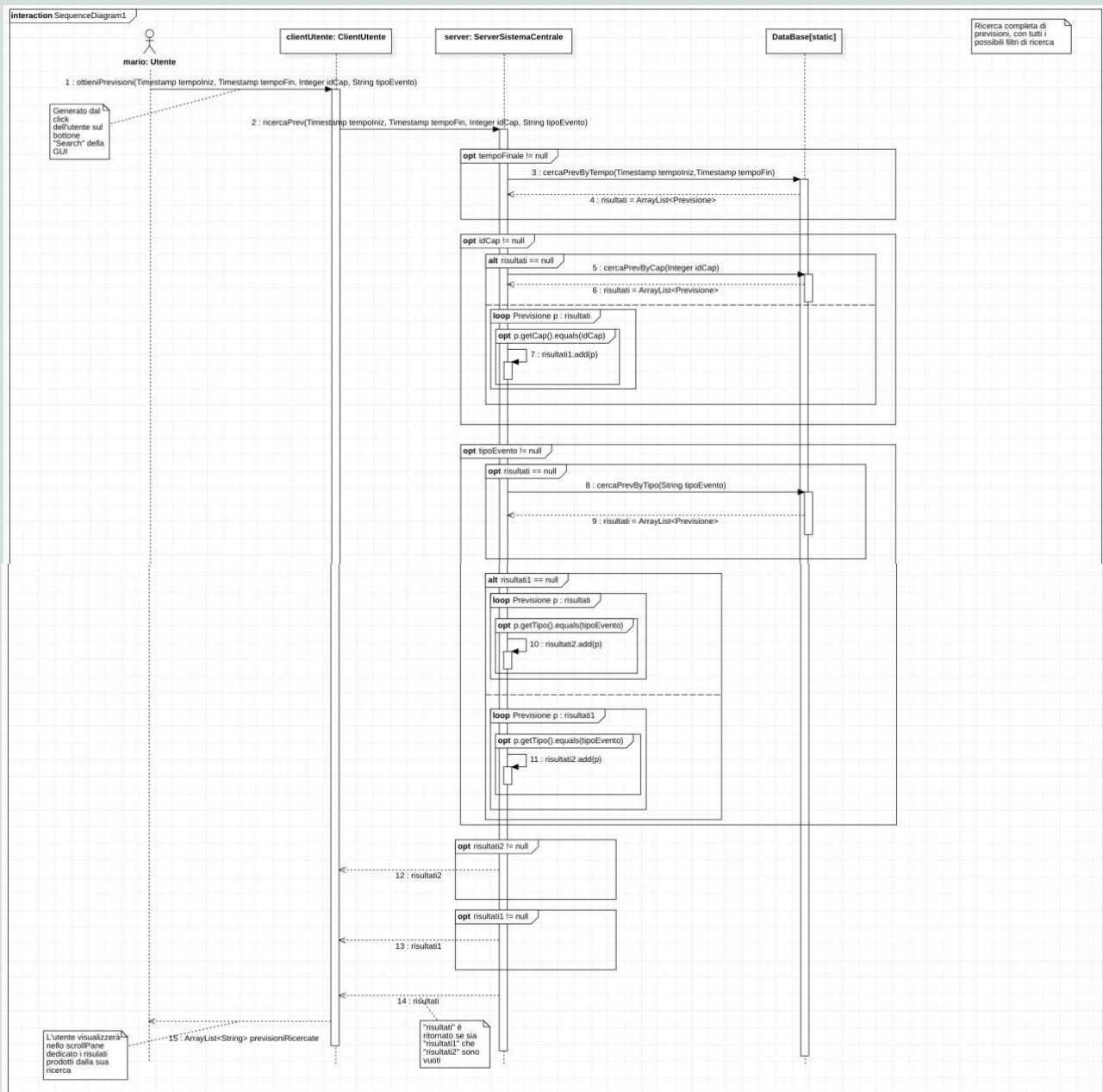


Lo state diagram descrive un processo attraverso dei grafi costituiti da nodi, che definiscono gli stati in cui si trova il sistema durante il processo, e da archi che descrivono le attività di passaggio tra uno stato e l'altro.

Nel caso in questione, il processo descritto riguarda la ricezione e il salvataggio di previsioni (meteo o terremoto).

Inizialmente il sistema è in stato di attesa di una nuova previsione; una volta ricevuta la previsione, il sistema attende l'analisi della presenza di possibili duplicati: se sono presenti duplicati, questi vengono rimossi e si attende il salvataggio della previsione, altrimenti si attende direttamente il salvataggio. Una volta salvata la previsione, il sistema torna nuovamente in stato di attesa.

## Sequence Diagram: ricerca completa di previsioni, con tutti i possibili filtri



In questo sequence diagram il processo descritto è la ricerca di previsioni da parte dell'utente, tenendo conto di tutte le possibili combinazioni di filtri che possono essere selezionati.

Tale processo ha inizio col click da parte dell'utente sul bottone di ricerca presente sull'interfaccia grafica; tale evento genera la chiamata di un metodo della classe client utente per ottenere le previsioni, la quale a sua volta chiama il sistema centrale comunicando i filtri inseriti dall'utente.

Il sistema centrale analizza quindi la richiesta ricevuta e, a seconda del valore dei filtri (che possono essere anche null se l'utente non ha selezionato alcun valore per un determinato filtro), decide quale procedimento seguire.

Il primo filtro che viene sempre analizzato è quello del tempo: se l'utente ha inserito una range temporale (data inizio e data fine), il sistema centrale chiama una funzione della classe database che estrapola dal database le previsioni previste per quel determinato range temporale e le salva in una lista chiama "risultati", altrimenti prosegue con l'analisi dei filtri successivi.

Il secondo filtro considerato è quindi quello relativo al cap: se non è stato inserito alcun valore per tale filtro, il sistema centrale prosegue senza compiere alcuna operazione, altrimenti deve tenere in considerazione due casi:

- se la lista "risultati" è vuota, ovvero non è stata fatta una ricerca per tempo, il sistema centrale chiama una funzione della classe database che estrapola dal database le previsioni previste per quel cap e le salva nella lista "risultati";
- se la lista "risultati" non è vuota, ovvero sono presenti nella lista delle previsioni ricavate dalla ricerca per tempo, il sistema centrale preleva da questa lista le previsioni che soddisfano anche la ricerca per cap e le salva in una nuova lista "risultati1".

A questo punto viene analizzato il terzo ed ultimo filtro, ovvero quello relativo al tipo di evento: se non è stato inserito alcun valore per tale filtro, il sistema centrale prosegue senza compiere alcuna operazione, altrimenti deve tenere in considerazione tre casi:

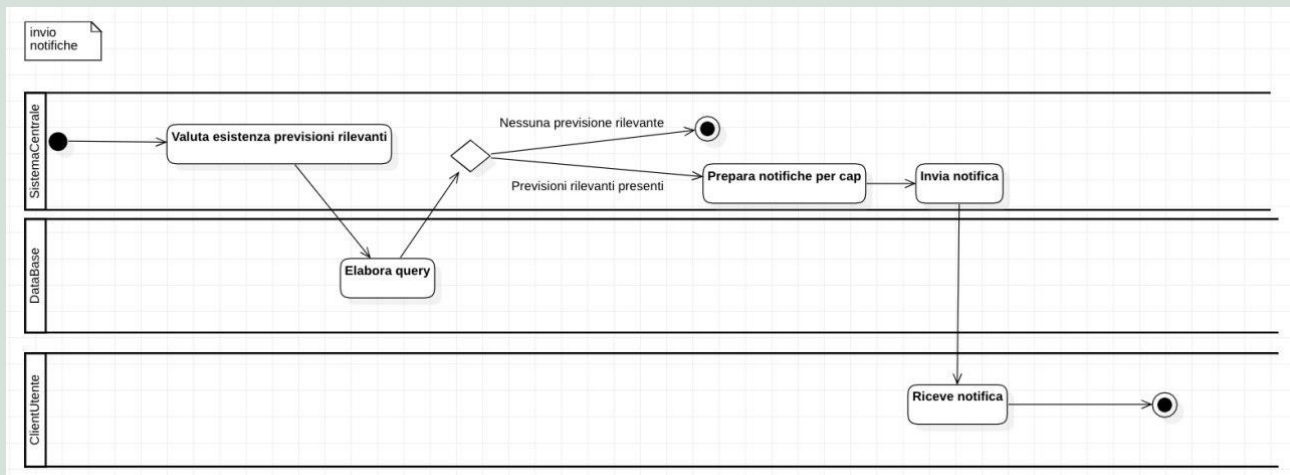
- se la lista "risultati" è vuota, significa che non è stata fatta una ricerca né per tempo né per cap e quindi il sistema centrale chiama una funzione della classe database che estrapola dal database le previsioni relative a quel tipo di evento e le salva nella lista "risultati";
- se la lista "risultati" non è vuota, ma lo è la lista "risultati1", significa che è stata già fatta una ricerca per tempo, ma non per cap; il sistema centrale quindi preleva dalla lista "risultati" solo le previsioni che soddisfano anche la ricerca per tipo e le salva nella lista "risultati1";
- se la lista "risultati" non è vuota e non lo è anche la lista "risultati1", significa che è stata fatta una ricerca sia per tempo che per cap; quindi il sistema centrale preleva dalla lista "risultati1" solo le previsioni che soddisfano anche la ricerca per tipo e le salva in una nuova lista "risultati2".

Considerati tutti i filtri, il sistema centrale analizza quindi le tre liste sopracitate partendo da "risultati2", proseguendo con "risultati1" e concludendo con "risultati": la prima lista che risulta non vuota viene ritornata alla classe client utente.

[Si noti che le liste vengono inizializzate a null.

La lista viene quindi mostrata all'utente tramite interfaccia grafica. Nel caso in cui essa sia vuota, viene mostrato un messaggio per comunicarlo.

## Activity Diagram: invio notifiche

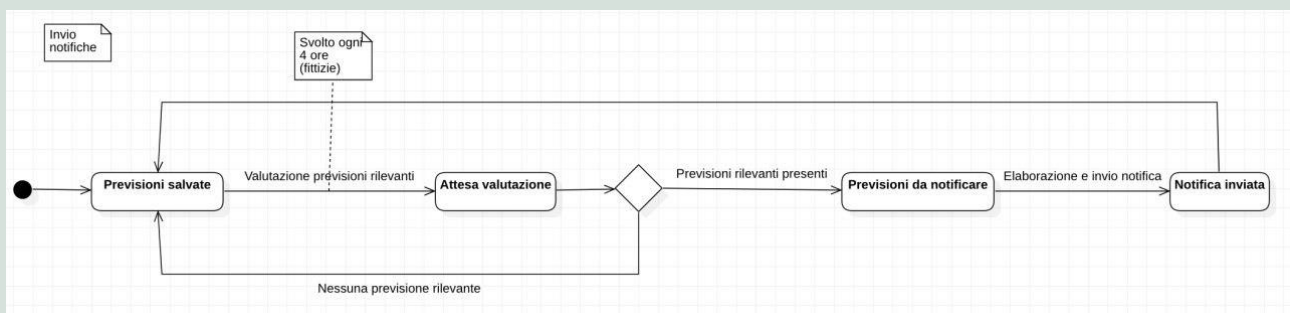


In tale activity diagram è stato descritto il processo che riguarda l'invio delle notifiche di allerta dal sistema centrale all'utente.

La prima attività viene svolta dal sistema centrale che valuta, per ogni cap, la presenza di previsioni con alto livello di gravità chiamando una funzione della classe database; tale classe interroga il database e ritorna il risultato al sistema centrale. Se il risultato non contiene nessuna previsione il processo termina, altrimenti il sistema centrale prepara le notifiche relative alle previsioni trovate e le inoltra alla classe client utente.

Quindi il processo termina.

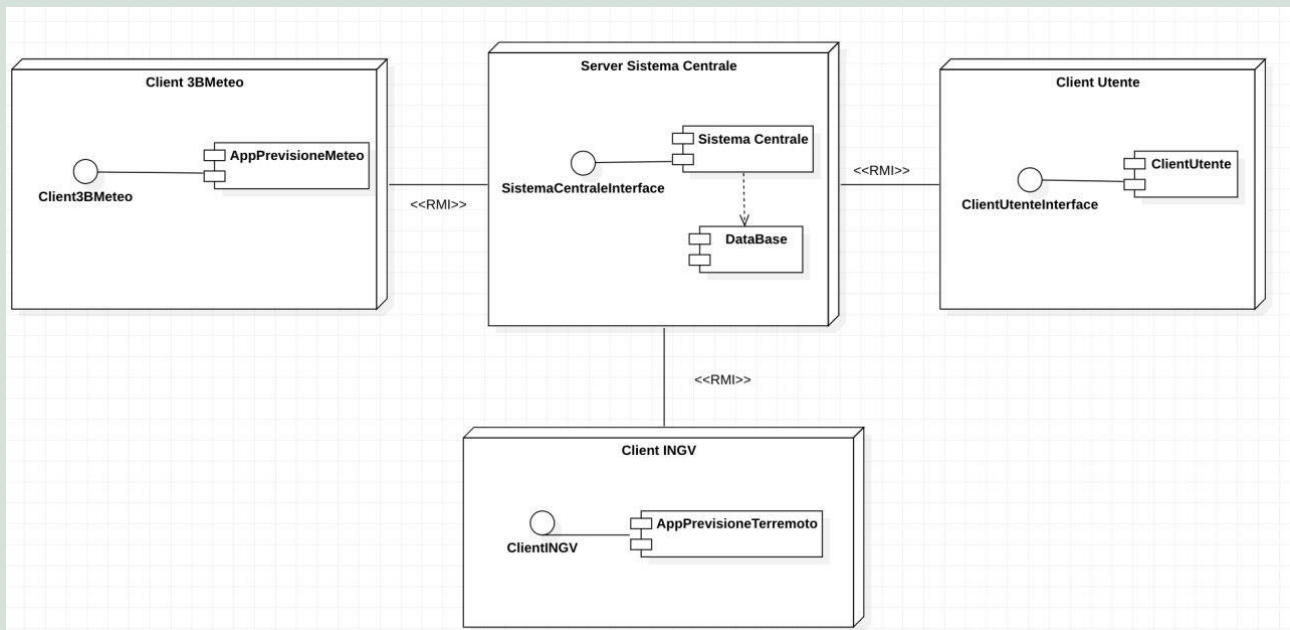
## State Diagram: invio notifiche



In tale state diagram il processo descritto riguarda l'invio delle notifiche.

Dopo che sono state salvate le previsioni, il sistema attende la valutazione di queste per estrapolare quelle di massima gravità (per ogni cap): se non sono presenti previsioni da allertare il sistema ritorna allo stato iniziale, altrimenti il sistema attende l'elaborazione delle notifiche e quindi il relativo invio, per poi riportarsi nello stato iniziale.

## Deployment Diagram

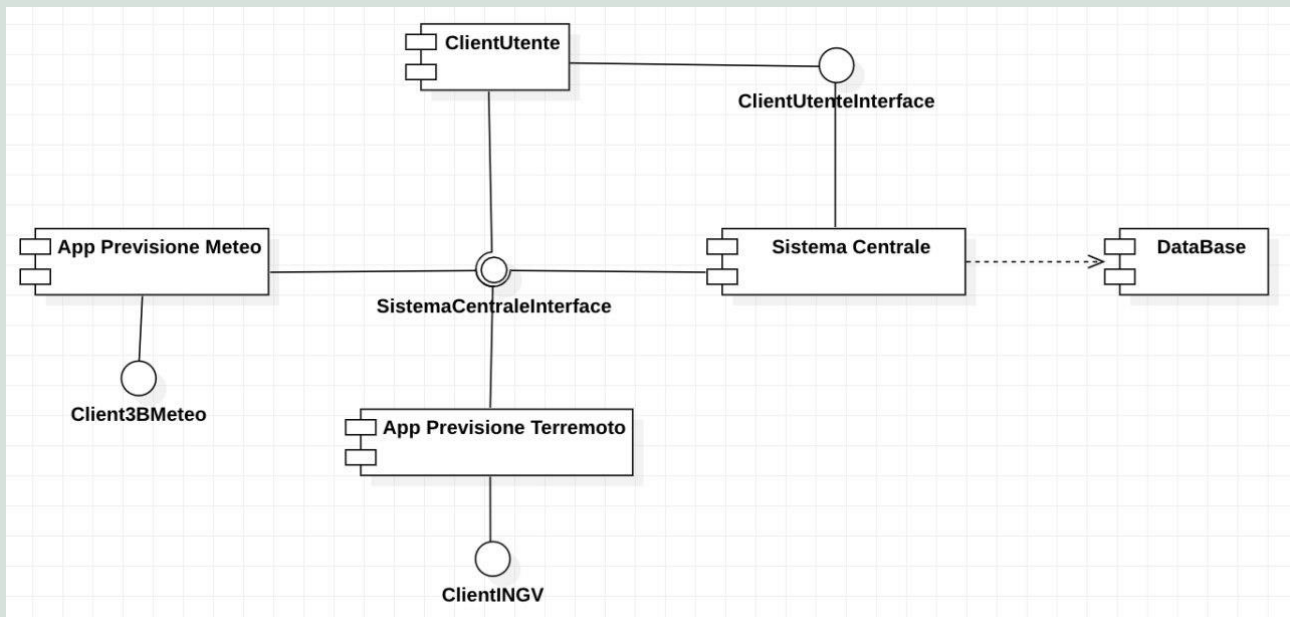


Il deployment diagram permette di descrivere le componenti hardware del sistema, specificando anche i loro componenti software e le loro relazioni.

Nel nostro caso le componenti hardware sono quattro:

- la componente relativa al sistema centrale che contiene il sistema centrale stesso e il database;
- le due componenti relative all'applicazione di previsione meteo e all'applicazione di previsione terremoto che contengono i rispettivi client adibiti alla comunicazione col sistema centrale (RMI);
- la componente relativa all'applicazione dell'utente che contiene il rispettivo client adibito alla comunicazione col sistema centrale (RMI).

## Component Diagram



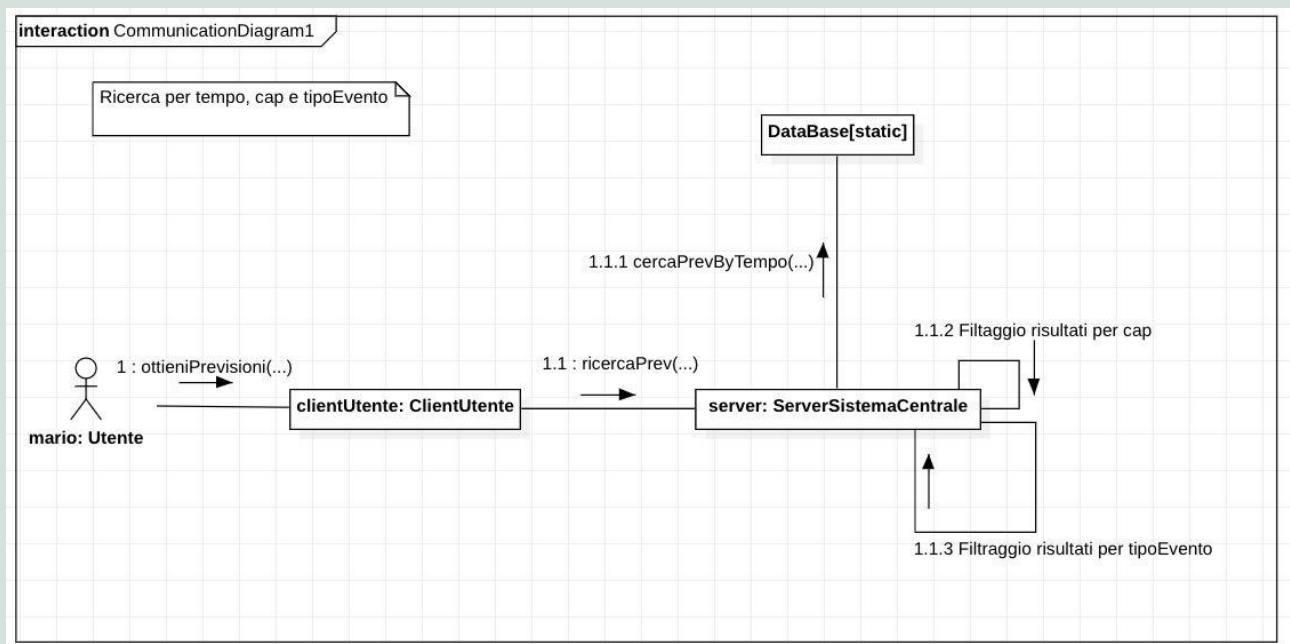
Il component diagram definisce le componenti software di un sistema con le relative interfacce, specificando le relazioni che sussistono tra esse.

Nel nostro caso le componenti software sono cinque:

- la componente del sistema centrale che implementa la sua interfaccia **SistemaCentraleInterface** e interagisce con il componente software **ClientUtente**, attraverso la relativa interfaccia, e con il database;
- la componente del database che comunica con il sistema centrale;
- le due componenti delle applicazioni di previsione (meteo e terremoto) che implementano le rispettive interfacce client e interagiscono con il sistema centrale tramite la sua interfaccia;
- la componente **ClientUtente** che implementa la sua interfaccia e interagisce con la componente software del sistema centrale tramite la relativa interfaccia.



## Collaboration Diagram: ricerca per tempo, cap e tipo evento



Il collaboration diagram mostra il flusso di interazioni e quindi le associazioni tra le istanze di diverse classi in particolari casi d'uso.

In tal caso, è stata descritta la ricerca di previsioni selezionando un valore per tutti i possibili filtri (tempo, cap e tipo).

L'interazione comincia da una istanza di Utente che richiede l'avvio della ricerca ad una istanza della classe ClientUtente attraverso l'invocazione di un metodo di questa; quest'ultima a sua volta richiama il metodo di ricerca della (unica) istanza del sistema centrale, il quale effettua inizialmente la ricerca per tempo interrogando il database e filtra successivamente i risultati ottenuti dalla query prima per cap e poi per tipo.

# IMPLEMENTAZIONE

# Commento Java

## Packages

Abbiamo diviso le classi del nostro programma in 8 pacchetti distinti, a seconda della categoria di appartenenza.

Il package principale è "serverSistemaCentrale" che contiene tutte le classi relative al sistema centrale stesso e al database:

- ConnectionPool**: gestisce il pool delle connessioni che vengono utilizzate dai vari processi del programma per collegarsi al database;

- DataBase**: contiene tutti i metodi che servono per elaborare query al database;

- ServerSistemaCentrale**: contiene le implementazioni di tutti i metodi che corrispondono alle funzionalità che deve svolgere il sistema centrale.

Vi sono poi i package relativi alle due applicazioni di previsioni meteo e terremoto.

Il primo, ovvero "trebmeteo", comprende:

- AppPrevisioneMeteo**: contiene le implementazioni dei metodi che corrispondono alle funzionalità dell'omonima applicazione;

- Client3BMeteo**: contiene il main che gestisce l'interazione tra l'applicazione e il sistema centrale via RMI e si occupa quindi di chiamare le funzioni necessarie (dell'applicazione o del sistema centrale).

Il secondo, ovvero "ingv", comprende analogamente:

- AppPrevisioneTerremoto**: contiene le implementazioni dei metodi che corrispondono alle funzionalità dell'omonima applicazione;

- ClientINGV**: contiene il main che gestisce l'interazione tra l'applicazione e il sistema centrale via RMI e si occupa quindi di chiamare le funzioni necessarie (dell'applicazione o del sistema centrale).

Un ulteriore package è "utente" che contiene le classi:

- Utente**: definisce le caratteristiche (attributi) di un utente e i relativi getter e setter;

- ClientUtente**: contiene l'implementazione dei metodi che corrispondono alle funzionalità dell'utente.

È presente poi il package "eventi" che comprende tutte le classi relative ai possibili eventi, ovvero:

- Evento**: definisce gli attributi di un evento e contiene i relativi getters e setters e il proprio costruttore; è la superclasse di tutte le classi appartenenti a questo package;

- Accadimento**: contiene il proprio costruttore ed è superclasse delle classi

AccadimentoMeteo e AccadimentoTerremoto;

- AccadimentoMeteo**: definisce l'attributo aggiuntivo "precipitazione", i relativi getter e setter e il proprio costruttore;

- AccadimentoTerremoto**: definisce l'attributo aggiuntivo "magnitudo", i relativi getter e setter e il proprio costruttore;

- Previsione**: definisce l'attributo aggiuntivo "livelloGravità", i relativi getter e setter e il proprio costruttore e contiene un metodo, richiamato dal costruttore, per la generazione casuale del tempo in cui si prevede la previsione; è superclasse delle classi PrevisioneMeteo e PrevisioneTerremoto;

- PrevisioneMeteo**: definisce l'attributo aggiuntivo "precipitazione", i relativi getter e setter e

il proprio costruttore;

-**PrevisioneTerremoto**: definisce l'attributo aggiuntivo "magnitudo", i relativi getter e setter e il proprio costruttore.

Abbiamo definito anche un package "condivise" che contiene le classi che possono essere utilizzate da altre classi appartenenti a più package diversi; tali classi sono:

-**SistemaCentraleInterface**: è l'interfaccia che contiene la dichiarazione dei metodi che vengono implementati dalla classe ServerSistemaCentrale;

-**ClientUtenteInterface**: è l'interfaccia che contiene la dichiarazione dei metodi che vengono implementati dalla classe ClientUtente;

-**ListaCap**: è una classe enumeration che contiene la lista dei cap a cui il sistema manda le notifiche di allerta, un metodo per prelevare randomicamente uno dei cap e un metodo per prelevare l'intera lista;

-**TipoEvento**: è una classe enumeration che contiene i possibili tipi di evento meteorologico che sono gestiti dal sistema, un metodo per prelevare un tipo di evento randomicamente e un metodo per prelevare l'intera lista;

-**Notifica**: definisce gli attributi di una notifica e contiene i relativi getters e setters e il proprio costruttore;

-**RandomGaussiano**: è la classe che contiene i metodi per la generazione di valori di precipitazione (per gli eventi di tipo meteo) o di magnitudo (per gli eventi di tipo terremoto) secondo una distribuzione gaussiana.

È presente inoltre il package "clientGUI" che contiene l'omonima classe che costituisce l'interfaccia grafica dell'applicazione dell'utente.

Infine, abbiamo creato un package "test" che contiene due classi di test JUnit [vedi paragrafo successivo].

## Test JUnit

Per testare il nostro programma, abbiamo creato due classi che verificano il corretto funzionamento dei metodi della classe database che gestiscono gli eventi.

Il primo test riguarda la gestione di una previsione meteo.

Viene quindi inizialmente generata una istanza della classe "PrevisioneMeteo", assegnandole determinati valori di test per ogni attributo, e si verifica che l'istanza creata contenga effettivamente tali valori scelti. L'istanza viene quindi inserita nel database.

Si verifica poi il corretto funzionamento di tutti i metodi possibili che riguardano la ricerca di previsioni; per farlo viene chiamato ogni metodo, passando i parametri necessari relativi alla nostra previsione di test, e si controlla per ciascuno che la previsione di test sia presente tra quelle trovate.

Lo stesso controllo viene effettuato sul metodo che ricerca le previsioni di massima gravità da notificare; abbiamo infatti assegnato un valore alto al parametro di test che riguarda il livello di gravità della previsione, in modo che questa risulti nella lista delle previsioni da allertare.

Infine, si cancella la previsione di test dal database utilizzando la funzione di cancellazione dei duplicati (per le previsioni) e si controlla che sia effettivamente stata eliminata utilizzando (arbitrariamente) la funzione che ricerca le previsioni tramite cap e verificando che essa non sia presente tra i risultati della ricerca.


Il secondo test è analogo, ma riguarda la gestione di un accadimento di tipo terremoto. Come nel primo caso, viene generata una istanza della classe "AccadimentoTerremoto", le vengono assegnati dei valori di test e si verifica che tali valori siano effettivamente associati all'istanza appena creata.

Si inserisce quindi l'accadimento di test nel database per poter verificare il corretto funzionamento dei metodi della classe database per la ricerca di accadimenti, così come fatto nel primo test.

Infine, si cancella l'accadimento dal database tramite la funzione di cancellazione dei duplicati (per gli accadimenti) e si controlla che sia stato effettivamente eliminato utilizzando (arbitrariamente) la funzione che ricerca gli accadimenti tramite cap e controllando che esso non sia presente tra i risultati della ricerca.

## Commento all'applicazione

### Login page



**BENVENUTO NELL'APP**  
*Protezione Civile*

**Inserisci il tuo username**  
Paolo Rossi

**Inserisci il cap di residenza**  
Seleziona cap

Login

L'applicazione all'avvio presenta la *Login Page* in cui è richiesto all'utente di inserire *Username* e *Cap*. Il *Cap* può essere scelto da un elenco finito di tutti i *Cap* esistenti. Nella nostra applicazione abbiamo deciso di inserire solo i *Cap* rappresentativi dei capoluoghi di provincia di 10 regioni scelte in modo casuale.

Una volta inseriti i campi richiesti è possibile cliccare il bottone "*Login*" per passare alla *Home Page*.

Se si scorre il cursore sui riquadri di inserimento compare un'*etichetta* che ricorda all'utente quale informazione inserire.



## Errore nella login page



Non è possibile accedere alla *Home Page* senza aver completato prima entrambi i campi.

Cliccare sul pulsante “*Login*” senza aver compilato entrambi, o solo uno dei due, produce un messaggio di errore che invita a completarli.

## Home page

Benvenuto Paolo Rossi (30100)  **Allarmi in evidenza** 

**Ricerca**

Data inizio  
24 mag 2020

Data fine

Tipo evento

Cap

☒ Previsioni  
☐ Accadimenti

Search

**Previsioni**

**Accadimenti/Storico**

Nell'immagine appena riportata si mostra la *Home Page*. Essa è composta da tre "box" che servono a contenere gli *allarmi in evidenza*, le *previsioni* e lo *storico degli accadimenti*.

Nella parte alta sinistra è riportato l'username e il relativo cap dell'utente che sta utilizzando l'applicazione.

A sinistra invece troviamo la sezione relativa ai filtri di ricerca dove è possibile selezionare *data*, *tipo evento*, *cap*. È inoltre possibile scegliere se ricercare una *previsione futura* o un *accadimento passato* (NB. Non è possibile selezionare entrambe le voci allo stesso tempo).

Infine, nella parte bassa sinistra è presente il pulsante di "*Search*" che una volta cliccato produrrà i risultati nei relativi box in accordo con i filtri inseriti.



## Ricerca di previsioni, filtri applicati: data e previsioni

The screenshot shows a web application window with a light blue header and a white body. The header contains a user greeting "Benvenuto Paolo Rossi (30100)", a yellow warning triangle icon, and the text "Allarmi in evidenza". A weather icon is in the top right corner. Below the header is a large red rectangular placeholder. The main content area is divided into two columns. The left column, titled "Ricerca", contains filters: "Data inizio" (24 mag 2020), "Data fine" (31 mag 2020), "Tipo evento" (a dropdown menu), and "Cap" (a dropdown menu). Below these are radio buttons for "Previsioni" (selected) and "Accadimenti", and a "Search" button. The right column, titled "Previsioni", displays a list of forecast events: "E' prevista Pioggia presso il cap 22030 in data 24 maggio 2020 alle ore 12:26 con precipitazione me...", "E' prevista Pioggia presso il cap 22030 in data 24 maggio 2020 alle ore 12:47 con precipitazione me...", "E' previsto un Terremoto presso il cap 50100 in data 29 maggio 2020 alle ore 20:46 con magnitudo 4", "E' previsto Vento presso il cap 23895 in data 24 maggio 2020 alle ore 18:27 con velocità media di 34", and "E' previsto un Terremoto presso il cap 30100 in data 27 maggio 2020 alle ore 01:39 con magnitudo 4". A scrollbar is visible on the right side of the list. Below the "Previsioni" section is a section titled "Accadimenti/Storico" with a large empty white box.

Benvenuto Paolo Rossi (30100) Allarmi in evidenza

**Ricerca**

Data inizio  
24 mag 2020

Data fine  
31 mag 2020

Tipo evento  
▼

Cap  
▼

☒ Previsioni  
☐ Accadimenti

Search

**Previsioni**

E' prevista Pioggia presso il cap 22030 in data 24 maggio 2020 alle ore 12:26 con precipitazione me  
E' prevista Pioggia presso il cap 22030 in data 24 maggio 2020 alle ore 12:47 con precipitazione me  
E' previsto un Terremoto presso il cap 50100 in data 29 maggio 2020 alle ore 20:46 con magnitudo 4  
E' previsto Vento presso il cap 23895 in data 24 maggio 2020 alle ore 18:27 con velocità media di 34  
E' previsto un Terremoto presso il cap 30100 in data 27 maggio 2020 alle ore 01:39 con magnitudo 4

**Accadimenti/Storico**

Completando i filtri relativi alla data e cliccando sul bottone "Search" viene generato il risultato della ricerca nel box corrispondente in base all' evento selezionato (nel nostro caso sono state ricercate delle previsioni).

Ricerca di accadimenti che non ha prodotto risultati, filtri applicati: Tipo evento, Cap e accadimenti



Benvenuto *Paolo Rossi (30100)*  **Allarmi in evidenza** 

**Ricerca**

Data inizio

Data fine

Tipo evento  
Pioggia

Cap  
87010

☐ Previsioni  
☒ Accadimenti

**Previsioni**

E' prevista Pioggia presso il cap 22030 in data 24 maggio 2020 alle ore 12:26 con precipitazione me  
E' prevista Pioggia presso il cap 22030 in data 24 maggio 2020 alle ore 12:47 con precipitazione me  
E' previsto un Terremoto presso il cap 50100 in data 29 maggio 2020 alle ore 20:46 con magnitudo 4  
E' previsto Vento presso il cap 23895 in data 24 maggio 2020 alle ore 18:27 con velocità media di 34  
E' previsto un Terremoto presso il cap 30100 in data 27 maggio 2020 alle ore 01:39 con magnitudo 4

**Accadimenti/Storico**

Nessun accadimento trovato corrispondente ai parametri di ricerca

Può capitare che la ricerca non produca risultati perché i filtri inseriti sono troppo restringenti e nel database non vengono trovate delle occorrenze relative ai filtri perché non presenti. In questo caso viene prodotto un messaggio che riporta *“Nessun accadimento trovato corrispondente ai parametri di ricerca”*.

Nel nostro caso questo messaggio compare nel box degli Accadimenti/Storico perché nei filtri di ricerca è stata selezionata la voce accadimento.

Ricerca di accadimenti, filtri applicati: Tipo evento, Cap e accadimenti

The screenshot shows a web application window with a light blue header. On the left, it says "Benvenuto Paolo Rossi (30100)". In the center, there is a yellow warning triangle icon and the text "Allarmi in evidenza". On the right, there is a weather icon showing a sun and a cloud. Below the header, there is a large red rectangular area. The main content area is divided into two columns. The left column is titled "Ricerca" and contains filters: "Data inizio" and "Data fine" (both with empty input fields and calendar icons), "Tipo evento" (a dropdown menu with "Terremoto" selected), and "Cap" (a dropdown menu with "50100" selected). Below these filters are two radio buttons: "Previsioni" (unselected) and "Accadimenti" (selected). At the bottom of the filters is a "Search" button. The right column is titled "Previsioni" and contains a list of forecast items: "E' prevista Pioggia presso il cap 22030 in data 24 maggio 2020 alle ore 12:26 con precipitazione me...", "E' prevista Pioggia presso il cap 22030 in data 24 maggio 2020 alle ore 12:47 con precipitazione me...", "E' previsto un Terremoto presso il cap 50100 in data 29 maggio 2020 alle ore 20:46 con magnitudo 4", "E' previsto Vento presso il cap 23895 in data 24 maggio 2020 alle ore 18:27 con velocità media di 34", and "E' previsto un Terremoto presso il cap 30100 in data 27 maggio 2020 alle ore 01:39 con magnitudo 4". Below the "Previsioni" section is a section titled "Accadimenti/Storico" which contains a single entry: "Si è verificato Terremoto presso il cap 50100 in data 24 maggio 2020 alle ore 14:50 con magnitudo 3.2".

Completando i filtri di Tipo evento e Cap e cliccando sul bottone "Search" viene generato il risultato della ricerca nel box corrispondente in base all' evento selezionato (nel nostro caso sono stati ricercati degli accadimenti).

## Ricezione notifica di allarme in evidenza

The screenshot shows a web application window with a light blue header and a white body. The header contains a user greeting 'Benvenuto Paolo Rossi (30100)' on the left, a yellow warning triangle icon in the center, and the title 'Allarmi in evidenza' on the right. A weather icon is also present in the top right corner. Below the header, a large red rectangular box contains the text: 'ATTENZIONE: si prevede Terremoto in data 31 maggio 2020 alle ore 24:16 con magnitudo 5.2 della scala Richter.' Below this box, the interface is divided into two main sections. On the left is a search panel titled 'Ricerca' with a magnifying glass icon. It includes input fields for 'Data inizio' (set to '24 mag 2020') and 'Data fine', a 'Tipo evento' dropdown menu, and a 'Cap' dropdown menu. At the bottom of this panel are two radio buttons: 'Previsioni' (selected) and 'Accadimenti', and a 'Search' button. On the right are two large white rectangular boxes. The top one is titled 'Previsioni' and the bottom one is titled 'Accadimenti/Storico'.

Quando viene registrata una previsione di gravità alta, viene prodotta una *Notifica* e inviata automaticamente all'utente senza che esso debba fare alcuna ricerca.

L'aggiornamento degli *allarmi in evidenza* avviene ogni minuto e vengono riportati all'interno del box rosso riservato proprio ad essi.

Se un'emergenza rientra, al successivo aggiornamento verrà rimossa la relativa notifica.

## Ricerca di previsioni, filtri applicati: Data, Tipo evento, Cap e previsioni

The screenshot shows a web application window with a light blue header and a white body. The header contains a user greeting "Benvenuto Paolo Rossi (30100)", a yellow warning triangle icon, and the text "Allarmi in evidenza". A weather icon is in the top right corner. A large red box in the header area contains the text: "ATTENZIONE: si prevede Terremoto in data 31 maggio 2020 alle ore 24:16 con magnitudo 5.2 della scala Richter." Below the header, the interface is divided into two main sections. On the left is the "Ricerca" (Search) section, which includes filters for "Data inizio" (24 mag 2020), "Data fine" (31 mag 2020), "Tipo evento" (Neve), and "Cap" (87010). It also has radio buttons for "Previsioni" (selected) and "Accadimenti", and a "Search" button. On the right is the "Previsioni" section, which displays a text box with the text: "E' prevista Neve presso il cap 87010 in data 30 maggio 2020 alle ore 13:30 con precipitazione media d". Below this is a horizontal scrollbar. At the bottom right is the "Accadimenti/Storico" section, which is currently empty.

Benvenuto Paolo Rossi (30100)  **Allarmi in evidenza** 

**ATTENZIONE:** si prevede Terremoto in data 31 maggio 2020 alle ore 24:16 con magnitudo 5.2 della scala Richter.

**Ricerca**

Data inizio  
24 mag 2020

Data fine  
31 mag 2020

Tipo evento  
Neve

Cap  
87010

☒ Previsioni  
☐ Accadimenti

Search

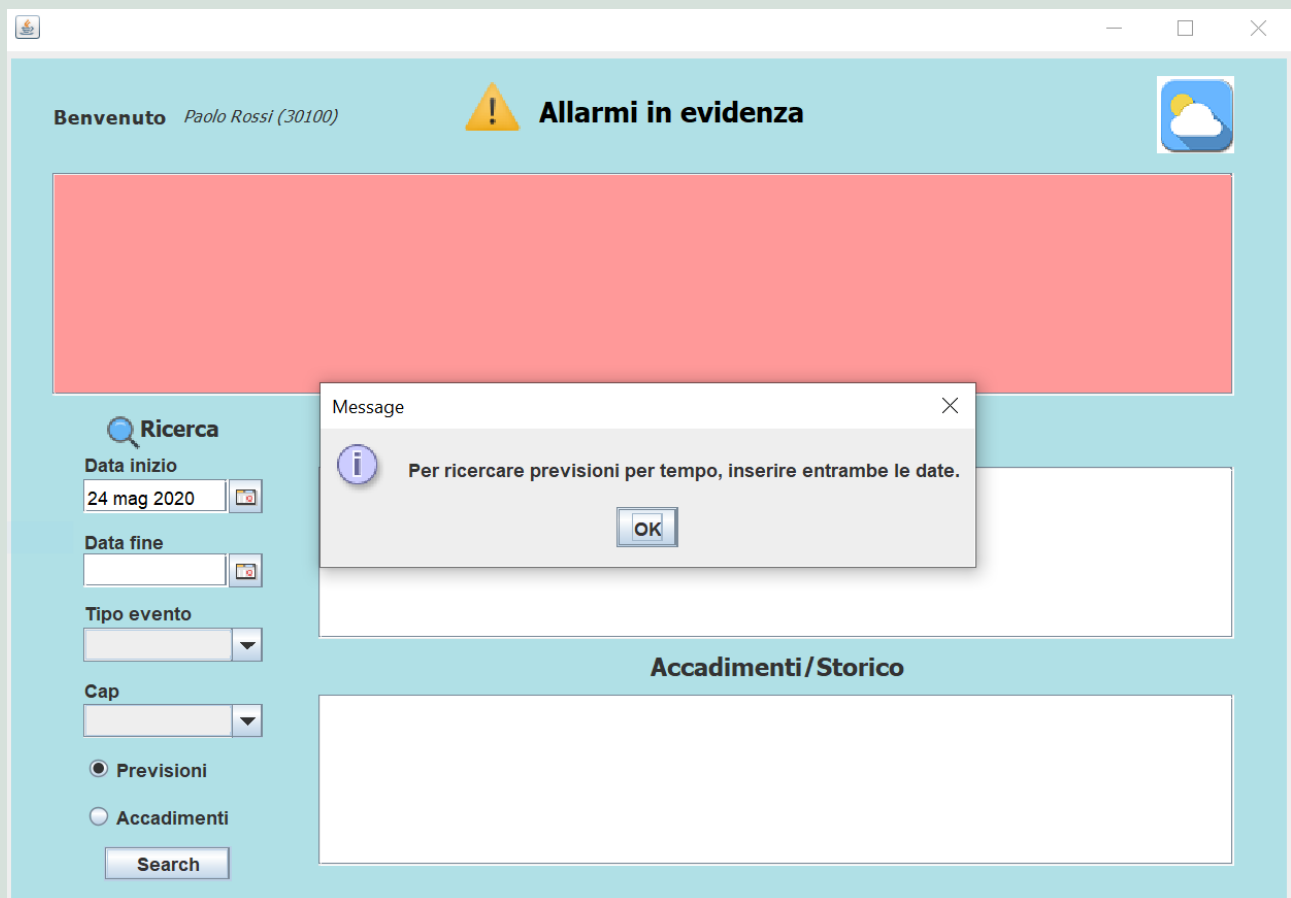
**Previsioni**

E' prevista Neve presso il cap 87010 in data 30 maggio 2020 alle ore 13:30 con precipitazione media d

**Accadimenti/Storico**

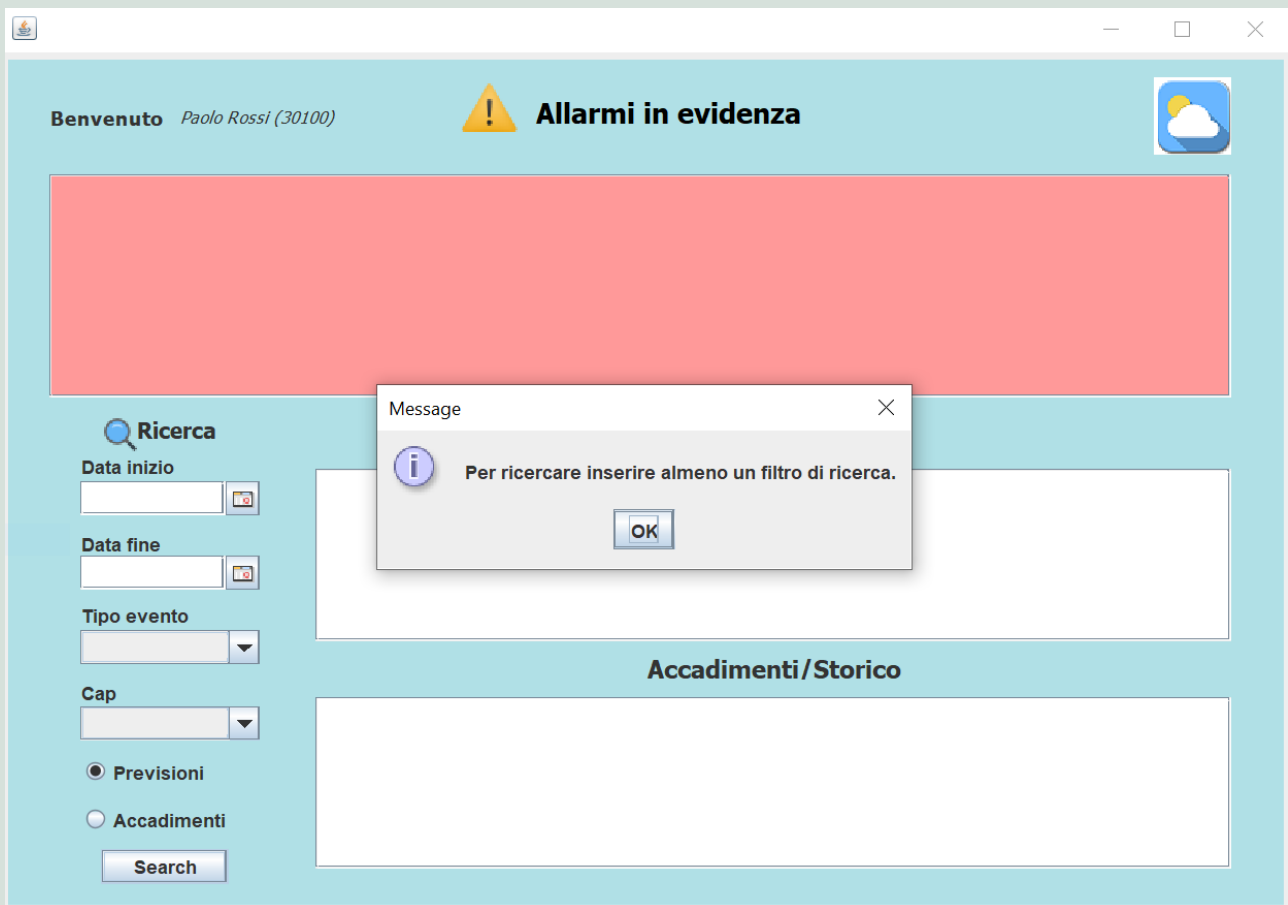
Completando tutti i filtri di data, tipo evento, cap e cliccando sul bottone “Search” viene generato il risultato della ricerca nel box corrispondente in base all’ evento selezionato (nel nostro caso sono state ricercate delle previsioni).

## Errore nella selezione dei filtri 1



Se si decide di applicare il filtro della data ma si completa solo uno dei due campi, al click sul bottone "Search" viene generato un messaggio di errore che invita l'utente a completarli entrambi.

## Errore nella selezione dei filtri 2



Se si clicca sul bottone “Search” senza aver selezionato alcun filtro viene generato un messaggio di errore che invita l’utente a completare almeno un filtro.

## Inserimento del filtro data

The screenshot shows a web application window with a light blue header. On the left, it says "Benvenuto Paolo Rossi (30100)". In the center, there is a yellow warning triangle icon and the text "Allarmi in evidenza". On the right, there is a weather icon. Below the header, there is a large red rectangular area. Underneath this, there is a search section labeled "Ricerca" with a magnifying glass icon. It contains two input fields: "Data inizio" with the value "24 mag 2020" and a calendar icon, and "Data fine" with an empty field and a calendar icon. To the right of the search section, there are two large white rectangular areas labeled "Previsioni" and "Accadimenti/Storico". A calendar popup is visible over the "Data inizio" field, showing the month of "maggio" and the year "2020". The calendar grid shows days from 18 to 22, with the 24th highlighted in red. A mouse cursor is pointing at the 24th. Below the calendar, there is a "Search" button.

Il campo del filtro data può essere compilato manualmente digitando la data da tastiera oppure scegliendola direttamente dal calendario a comparsa come mostrato nell'immagine appena riportata.

Si noti che vengono mostrate solo le date selezionabili, le quali cambiano in base al tipo di evento selezionato: previsione o accadimento.

Per le previsioni non si possono selezionare date antecedenti alla data corrente, mentre per gli accadimenti non è possibile selezionare date successive a quella odierna.



## Inserimento del filtro Tipo evento

The screenshot shows a web application window with a light blue header. On the left, it says "Benvenuto Paolo Rossi (30100)". In the center, there is a yellow warning triangle icon followed by the text "Allarmi in evidenza". On the right, there is a weather icon showing a sun and a cloud. Below the header, there is a large pink rectangular area. Underneath this, the interface is divided into two main sections. The left section is titled "Ricerca" and contains three input fields: "Data inizio" with the value "24 mag 2020", "Data fine", and "Tipo evento". The "Tipo evento" field is a dropdown menu that is currently open, showing a list of options: "Pioggia", "Neve", "Grandine", "Vento", and "Terremoto". A mouse cursor is pointing at the "Grandine" option. Below the dropdown menu is a "Search" button. The right section is titled "Previsioni" and contains a large empty white box. Below this, there is another section titled "Accadimenti/Storico" which also contains a large empty white box.

Il campo del filtro tipo evento viene compilato scegliendo un'opzione tra quelle proposte nel menu a tendina.

È presente anche l'opzione nulla che permette di cancellare il campo a seguito di un eventuale inserimento precedente.

Lo stesso metodo di selezione avviene per la compilazione del campo cap.

## Database: tabella delle previsioni

	idprevisione	cap	tipo	sorgente	tempo	livellogravità	precipitazione	magnitudo
►	447	22030	Neve	3bMeteo	2020-05-29 18:17:10	9	1.5	NULL
	448	20100	Pioggia	3bMeteo	2020-05-30 23:52:09	5	22.7	NULL
	449	90121	Neve	3bMeteo	2020-05-31 04:32:33	9	1.5	NULL
	450	80010	Vento	3bMeteo	2020-05-30 01:04:06	4	31.3	NULL
	451	30100	Terremoto	Istituto Nazionale di Geofisica e Vulcanologia	2020-05-31 00:16:50	6	NULL	5.2
	452	22030	Terremoto	Istituto Nazionale di Geofisica e Vulcanologia	2020-05-26 22:47:50	2	NULL	1.9
	453	80010	Terremoto	Istituto Nazionale di Geofisica e Vulcanologia	2020-05-31 06:29:10	4	NULL	3.7
	454	20100	Terremoto	Istituto Nazionale di Geofisica e Vulcanologia	2020-05-25 05:15:44	6	NULL	5
	455	22030	Neve	3bMeteo	2020-05-27 12:11:01	10	2.1	NULL
	456	50100	Neve	3bMeteo	2020-05-31 18:01:41	8	1.3	NULL
	457	10121	Vento	3bMeteo	2020-05-24 23:07:03	3	27.1	NULL
	458	23895	Pioggia	3bMeteo	2020-05-29 15:39:09	3	9.8	NULL

Questa appena riportata è la tabella che mostra l'elenco delle *previsioni*. Per ognuna vengono registrati *id*, *cap*, *tipo*, *sorgente*, *tempo*, *livello gravità* e *precipitazione* o *magnitudo* a seconda del tipo di evento (evento meteo o terremoto).

Il database viene aggiornato ogni volta che riceve una nuova previsione dalle applicazioni di previsione.

## Database: tabella degli accadimenti

	idaccadimento	cap	tipo	sorgente	tempo	precipitazione	magnitudo
►	96	90121	Vento	3bMeteo	2020-05-24 21:41:02	10.6	NULL
	97	90121	Terremoto	Istituto Nazionale di Geofisica e Vulcanologia	2020-05-24 21:41:10	NULL	3.6
	100	87010	Pioggia	3bMeteo	2020-05-24 21:41:53	26.7	NULL
	101	22020	Terremoto	Istituto Nazionale di Geofisica e Vulcanologia	2020-05-24 21:42:10	NULL	4.9
	102	23895	Grandine	3bMeteo	2020-05-24 21:42:18	11.5	NULL
	103	10121	Terremoto	Istituto Nazionale di Geofisica e Vulcanologia	2020-05-24 21:42:40	NULL	2.7
	104	22020	Grandine	3bMeteo	2020-05-24 21:42:43	16.7	NULL

Questa appena riportata è la tabella che mostra l'elenco degli accadimenti. Per ognuno vengono registrati *id*, *cap*, *tipo*, *sorgente*, *tempo* e *precipitazione* o *magnitudo* a seconda del tipo di evento (evento meteo o terremoto).

Il database viene aggiornato ogni volta che una previsione si verifica.