

# Deliverable 3 – CBAM

---

DANILO DELL'ORCO 0300229

# Roadmap

Introduzione

Definizione decisione e alternative

Attributi di qualità

Valutazione alternative

Valutazione rischio

Calcolo beneficio alternative

Valutazione costo alternative

Calcolo desiderabilità alternative

Classifica

Conclusioni

# Introduzione (1/2)

---

- Si vuole sviluppare un software in ambiente Java per la *gestione di un'isola ecologica*.
- Tale applicazione, chiamata *BeEcological*, deve offrire diverse funzionalità ad utenti e proprietari di un centro.
  - Un utente può effettuare una prenotazione verso un centro
  - Un proprietario può confermare o declinare una prenotazione

# Introduzione (2/2)

---

- E' necessario definire l'architettura software del sistema.
  - Scelta della struttura del sistema e dei componenti software da utilizzare nel progetto.
- Si vuole applicare a tale scopo **Cost Benefit Analysis Method**
  - Tecnica decisionale che permette di selezionare la *migliore* tra *diverse alternative*, valutando per ognuna **costi** e **benefici**

# Decisione e Alternative

---

- Per l'implementazione di BeEcological è necessario l'utilizzo di un **DBMS**, al fine di mantenere la *persistenza* dei dati relativi agli utenti, ai centri, e a tutte le richieste di prenotazione effettuate.
- Sono state individuate tre possibili alternative: **PostgreSQL, MySQL, MongoDB**.

# Attributi di Qualità

---

1. Performance
2. Security
3. Modifiability
4. Availability
5. Interoperability
6. Integrability

# Attributi di Qualità - Descrizione

---

- Le **performance** indicano la velocità del DBMS nelle operazioni di lettura/scrittura delle tabelle.
- La **security** specifica quanto il database sia privo di vulnerabilità.
- La **modifiability** indica la facilità con cui è possibile apportare modifiche all'interno del DB.
- L' **availability** indica per quanto tempo il database è effettivamente accessibile, rispetto al tempo totale in cui è attivo
- L' **interoperability** indica quanto facilmente il DBMS può comunicare servizi esterni
- L' **integrability** indica quanto facilmente il DBMS può essere integrato all'interno di altri sistemi

# Attributi di Qualità - Contributo

---

1. Performance	20
2. Security	14
3. Modifiability	22
4. Availability	9
5. Interoperability	15
6. Integrability	20



# Valutazione Alternative - PostgreSQL

---

1. Performance	0.0	✓ DBMS Altamente scalabile, capace di gestire anche terabytes di dati.
2. Security	0.5	✓ Offre supporto nativo a JSON.
3. Modifiability	-0.3	✓ Offre moltissime interfacce e funzioni predefinite.
4. Availability	0.9	
5. Interoperability	0.2	✗ Configurazione iniziale complessa.
6. Integrability	0.5	✗ Velocità di lettura non ottimale quando il carico di lavoro è troppo pesante
		✗ Documentazione carente

# Valutazione Alternative - MySQL

---

1. Performance	0.2	0.0	✓ Facilità di configurazione e utilizzo
2. Security	0.0	0.5	✓ Facilità di integrazione con altri database e linguaggi
3. Modifiability	0.9	-0.3	✓ Ampia documentazione disponibile
4. Availability	-0.3	0.9	
5. Interoperability	0.7	0.2	✗ Non presenta supporto nativo ad alcune funzioni come <i>incremental backups</i>
6. Integrability	0.8	0.5	✗ Non offre supporto nativo per XML o OLAP ✗ Non offre una buona scalabilità orizzontale rispetto al numero di tabelle

# Valutazione Alternative- MongoDB

---

1. Performance	0.9	✓ Offre prestazioni elevate in lettura/scrittura
2. Security	-0.2	✓ Offre supporto nativo a JSON e ad altri documenti NoSQL.
3. Modifiability	0.4	✓ Molto semplice da utilizzare
4. Availability	0.7	
5. Interoperability	0.3	✗ Non utilizza SQL come <i>query language</i> , sono necessari dei tool per tradurre le query da SQL a MongoDB.
6. Integrability	0.8	✗ Configurazione iniziale complessa. ✗ La configurazione di default presenta diverse vulnerabilità

# Rischio

---

- PostgreSQL: 0.1
  - PostgreSQL presenta un *basso rischio* dovuto alla mancanza di documentazione presente in rete. Durante il processo di sviluppo potrebbe rivelarsi particolarmente laboriosa l'implementazione delle funzionalità più complesse.
- MySQL: 0
  - MySQL non presenta *alcun rischio*, in quanto il team di sviluppo ha familiarità con tale DBMS. L'ampia documentazione presente in rete garantisce inoltre un maggiore supporto allo sviluppo di funzionalità più elaborate.
- MongoDB: 0.3
  - MongoDB presenta un *rischio* in quanto non utilizza *query SQL* come la maggior parte dei DBMS. Potrebbe quindi rivelarsi molto più complessa l'interazione con il sistema, rendendo necessario l'utilizzo di un tool esterno per la conversione delle query.

# Beneficio

---

$$Benefit(AS_i) = \sum_{j=1}^7 ((Cont_{ij} * QAscore_j)) * |Risk_i - 1|$$

$$Benefit(PostgreSQL) = (20 * 0.0 + 14 * (0.5) + 22 * (-0.3) + 9 * 0.9 + 15 * 0.2 + 20 * 0.5) * (1 - 0.1)$$

$$Benefit(PostgreSQL) = \mathbf{19.35}$$

$$Benefit(MySQL) = (20 * 0.2 + 14 * 0.0 + 22 * 0.9 + 9 * (-0.3) + 15 * 0.7 + 20 * 0.8) * (1)$$

$$Benefit(MySQL) = \mathbf{47.6}$$

$$Benefit(MongoDB) = (20 * 0.9 + 14 * (-0.2) + 22 * 0.4 + 9 * 0.7 + 15 * 0.3 + 20 * 0.8) * (1 - 0.3)$$

$$Benefit(MongoDB) = \mathbf{35.56}$$

# Costo

---

- Tutti e tre i software presi in considerazione sono disponibili con *licenza gratuita open source*
- Il costo delle tre alternative è considerato come unitario
  - $Cost(PostgreSQL) = 1$
  - $Cost(MySQL) = 1$
  - $Cost(MongoDB) = 1$

# Desiderabilità

---

$$Desiderabiliy(AS_i) = \frac{Benefit(AS_i)}{Cost(AS_i)}$$

$$Desiderabiliy(PostgreSQL) = \frac{Benefit(PostgreSQL)}{Cost(PostgreSQL)} = 19.35$$

$$Desiderabiliy(MySQL) = \frac{Benefit(MySQL)}{Cost(MySQL)} = 47.6$$

$$Desiderabiliy(MongoDB) = \frac{Benefit(MongoDB)}{Cost(MongoDB)} = 35.56$$

# Classifica

---

1. MySQL
2. MongoDB
3. PostgreSQL



# Conclusioni

---

- MySQL rappresenta la migliore scelta tra i DBMS considerati, in quanto è quello che offre un maggiore score di desiderabilità.
  - $Desiderability(MySQL) = 47.6$
- MongoDB risulta essere una scelta accettabile, avendo raggiunto uno score di desiderabilità non troppo inferiore a MySQL.
  - $Desiderability(MongoDB) = 35.56$
  - Inferiore del 27% rispetto a MySQL
- PostgreSQL rappresenta invece una alternativa decisamente *sconsigliabile* rispetto alle altre due possibilità
  - $Desiderability(PostgreSQL) = 19.35$
  - Inferiore del 45% rispetto a MongoDB
  - Inferiore del 60% rispetto a MySQL

# Riferimenti

---

- [1] <https://www.keycdn.com/blog/popular-databases>
- [2] <https://acodez.in/list-popular-databases>
- [3] <https://www.xplenty.com/blog/which-database/>
- [4] <https://stackoverflow.com/questions/6475228/postgresqls-security-compared-to-mysql-etc>
- [5] <https://www.enterprisedb.com/blog/postgresql-vs-mysql-360-degree-comparison-syntax-performance-scalability-and-features>
- [6] <https://db-engines.com/en/system/MongoDB%3BMySQL%3BPostgreSQL>