Documentazione AFU

Crea un programma per la gestione di un contocorrente. Il programma deve essere in grado di effettuare versamenti, prelievi, di restituire il saldo e di estrarre la lista degli ultimi 5 movimenti.

Documentazione ATE

Controller

GET localhost:8080/api/v1/movimento/{iban}: restituirà la lista degli ultimi 5 movimenti

GET localhost:8080/api/v1/conto/{iban}: restituisce il saldo dell'utente

PUT localhost:8080/api/v1/conto/{iban}{importo}: Aumenta il saldo dell'importo specificato del conto relativo all'iban per un versamento

PUT localhost:8080/api/v1/conto/{iban}{importo} prelievo(int x): diminuisce il saldo dell'importo specificato del conto relativo all'iban per un prelievo

PUT localhost:8080/api/v1/conto/{iban}: modifica dati di un conto corrente
PUT localhost:8080/api/v1/movimento/{id}: modifica dati di un movimento
PUT localhost:8080/api/v1/persona/{cf}: modifica dati di una persona

POST localhost:8080/api/v1/conto/{iban}: aggiunge un conto corrente
POST localhost:8080/api/v1/movimento/{id}: aggiunge un movimento
POST localhost:8080/api/v1/persona/{cf}: aggiunge una persona

DELETE localhost:8080/api/v1/conto/{iban}: cancella un conto corrente

DELETE localhost:8080/api/v1/movimento/{id}: cancella un movimento

DELETE localhost:8080/api/v1/persona/{cf}: cancella una persona

ContoService

getSaldo: ritorna il saldo effetuando la chiamata getById() da ContoRepository
versamento: versamento di x€, aumenta il saldo di x chiama getById() da ContoRepository

prelievo: diminuisce il saldo di x chiamando updateById da ContoRepository

versamento: aumenta il saldo di x chiamando updateById da ContoRepository

Ogni versamento e prelievo sarà registrato come ‘Movimento’

MovimentoService

getMovimenti: prende la query dal repository, filtra gli ultimi 5 movimenti e li manda al Controller

modificaMovimento:modifica un movimento relativo ad un conto corrrente chiamando il metodo UpdateByID da MovimentoRepository
rimuoviMovimento: modifica un movimento relativo ad un conto chiamando deleteById da MovimentoRepository

PersonaService

aggiungiPersona: chiama il metodo save da PersonaRepository
modificaPersona: chiama il metodo updateByID da PersonaRepositiry
rimuoviPersona: chiama il metodo deleteByID da PesonaRepository

ContoRepository

Interfaccia che estende JPARepository

save: crea un nuovo conto corrente associato ad una persona con saldo iniziale a 0

updateById: inserire nuovo saldo, per aumentare o diminuire il saldo
deleteByID: cancella un conto corrente dal Database

PersonaRepository

save: aggiunge una persona al Database
updateById: modifica i dati relativi ad una persona
deleteById: cancella la persona dal Database

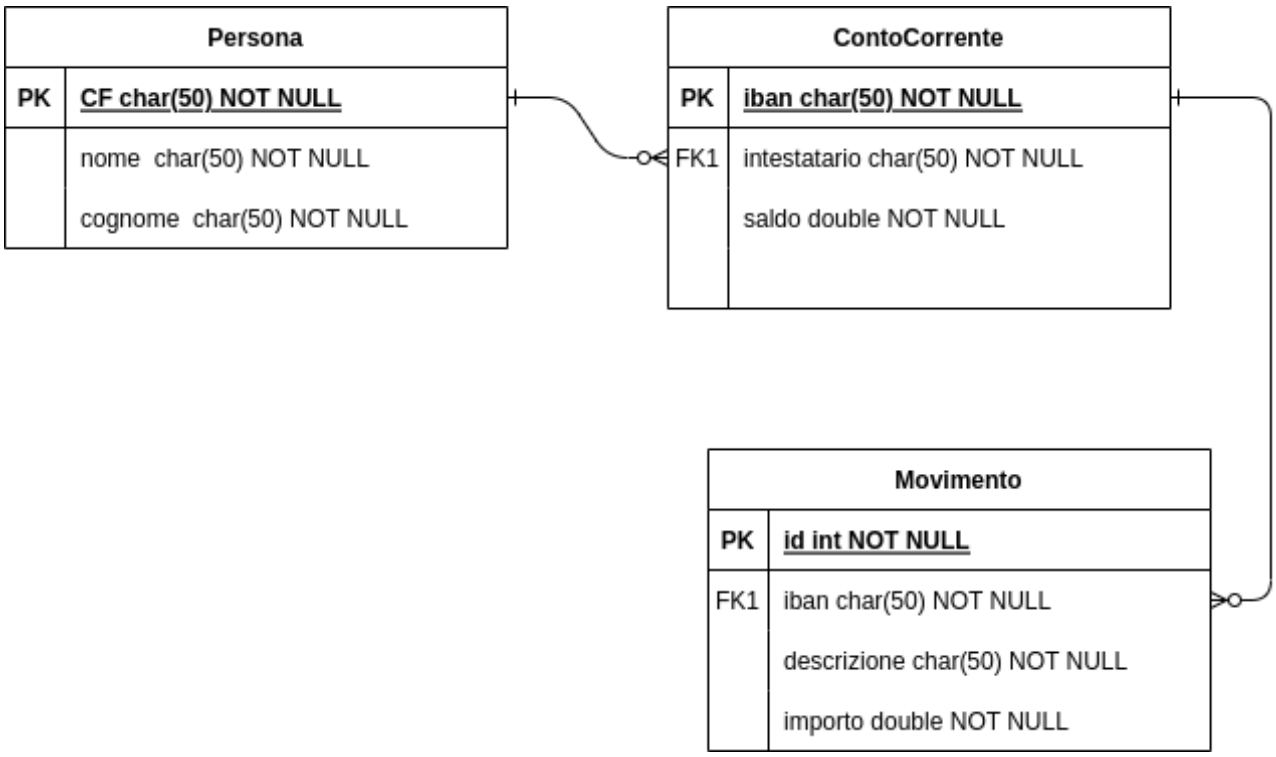
MovimentoRepository

save: salva un nuovo movimento relativo ad un conto corrente inserendo la descrizione del movimento effettuato e l’importo
updateById: modifica un movimento

deleteById: cancella un movimento

Le classi Persona, ContoCorrente e Movimento conterranno i dati descritti nel DB e le relative associazioni.

DIAGRAMMA E-R



DATABASE

```
CREATE TABLE IF NOT EXISTS `persona` (
  `cf` varchar(50) NOT NULL,
  `nome` varchar(50) NOT NULL,
  `cognome` varchar(50) NOT NULL,
  PRIMARY KEY (`cf`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;


CREATE TABLE IF NOT EXISTS `movimento` (
```

```
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `iban` varchar(50) NOT NULL,
  `descrizione` varchar(50) NOT NULL,
  `importo` double NOT NULL,
  PRIMARY KEY (`id`),
  KEY `FK_movimento_contocorrente` (`iban`),
  CONSTRAINT `FK_movimento_contocorrente` FOREIGN KEY (`iban`)
REFERENCES `contocorrente` (`iban`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

CREATE TABLE IF NOT EXISTS `persona` (
  `cf` varchar(50) NOT NULL,
  `nome` varchar(50) NOT NULL,
  `cognome` varchar(50) NOT NULL,
  PRIMARY KEY (`cf`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
```

STIME

Creazione Database: 1h
ContoRepository: 45m
PersonaRepository: 45m
MovimentoRepository: 45m
ContoService: 30m
PersonaService: 30m
MovimentoService: 30m
PersonaController: 45m
MovimentoController: 45m
ContoController: 45m
Persona: 30m
Conto: 30m
Movimento: 30m

Tempo totale: 8h 30m