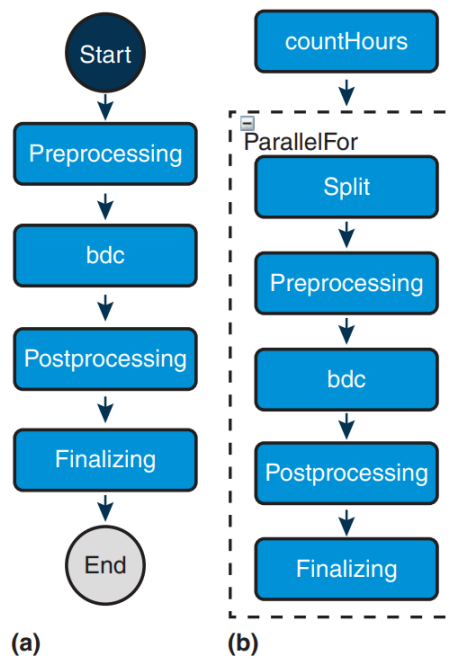


Progetto

🕒 Created	@March 28, 2025 7:32 PM
📁 Class	Esame SDCC

▼ Microservizio Analisi dati remoti

- usiamo AWS lambda per le singole funzioni e le componiamo con step function
- figura a per processamento streaming, figura b per processamento batch
 - ~~preprocessing~~ → ~~filtra i dati di input per rimuovere il rumore~~ → ~~es_ filtro di una forma d'onda, normalizzazione dei dati in input~~
 - bdc → **estrazione delle features** → se algoritmo ml già implementato estraggo dai dati di input solo quelli necessari all'algoritmo e se devo aggiusto il formato
 - post-processing → vero e proprio **algoritmo ml**
 - **finalizing** →
 - se presente anomalia chiamata a sns per allert al medico
 - se non presenta anomalie → *facciamo mandare comunque i dati al db che può visualizzare il medico nel caso volesse esaminarli?* → valore = VALORI NELLA NORMA: ...
 - se presenta anomalie →
 - i risultati con anomalie vengono salvati in un file chiave-valore nel db del microservizio gestione cartella clinica → valore = ANOMALIA: ...
 - a fine giornata il medico userà tali valori per redigere un resoconto della anomalia presentata dal paziente in tale giorno



- comunicazione dati per streaming con IoT core →
 - simuliamo IoT con script Python
- comunicazione dati batch →
 - caricamento manuale di un file nel bucket s3 → s3 tramite trigger invia alle funzioni lambda di analisi i dati caricati

▼ Microservizio Gestione cartella clinica

vedi finalizing sopra

▼ Microservizio Visualizzazione cartella clinica (paziente)

sorta di diario del paziente, giorno x → valori nella norma/ descrizione dell'anomalia presentata

con suo db separato in cui il medico scrive il report della giornata in caso di anomalie

▼ Microservizio Autenticazione

cognito

▼ Microservizio Gestione medici (amministratore)

registrazione del medico nel db per autenticazione

▼ **Microservizio Gestione pazienti (medico)**

registrazione del paziente nel db per autenticazione

- Per nostro modello su SageMaker necessario Tensorflow 2.2
- Guardare se dati in input necessitano della moltiplicazione come spiegato da chat o già normalizzati per il modello
- nei dati in input avremo un array contenente 328 array, uno per ogni ecg, che contengono 4096 righe che corrispondono ai campioni rilevati, ognuna delle quali sarà formata da 12 valori, una per ogni canale dell'ecg (il cvs è stato appiattito a 2 dimensioni, probabilmente numero di colonne pari a 4096×12 o 4096×328 o 12×328 → da controllare)
- Dispositivo → Biosensori clinici tipo Bitalino, Shimmer3 ECG, oppure custom board con ADC + multiplexer → Trasmissione:
 - Invia blocchi via MQTT/HTTP ogni 10 secondi
 - Encoding binario o compressione (es. FlatBuffers)

Quindi in questo caso i dati in input li simuliamo a partire da quelli che sono stati usati per addestrare il modello → li inviamo usando MQTT (direttamente lambda/passando per API → opterei per la seconda per maggiore manutenibilità rispetto a come viene effettuata la predizione) → a questo punto serve davvero IoT core? Da ricontrollare lo schema che avevano sui progetti che avevamo visto inizialmente)

- Deployment serverless via → AWS Lambda con GPU inference (con SageMaker)
- Un'infrastruttura di rilevazione e analisi real-time di un ECG 12-canali "serverless" può essere alla base di molteplici casi d'uso clinici e commerciali. Ecco i principali:

1. Telecardiologia e monitoraggio remoto

- **Pazienti cronici** (scompenso, aritmie note): invio continuo o on-demand di tracciati ECG a distanza verso centri specializzati per aggiustamenti terapeutici senza visita in ospedale.
- **Post-dimissione ospedaliera**: riduce i ricoveri ripetuti identificando precocemente recidive o complicanze.

2. Diagnosi e screening precoce

- **Screening a domicilio** di fibrillazione atriale (AFib) o blocchi di branca asintomatici, con avviso automatico al cardiologo.

- **Check-up aziendali**, farmacie o cliniche mobili: acquisizione su tablet/tablet-embedded, analisi automatica e reportistica immediata.

3. Emergenza e primo soccorso

- **Ambulanze e automediche**: invio in tempo reale dell'ECG al PS più vicino per triage pre-ospedaliero (infarto miocardico, STEMI-alert).
- **Dispositivi indossabili per atleti**: in endurance (ciclismo, triathlon) o sport di squadra, per intercettare aritmie pericolose sotto sforzo.

4. Telemedicina aziendale e insurance

- **Programmi di wellness aziendale**: integrazione in app di employee-health, con advising medico automatico in caso di anomalie.
- **Assicurazioni sanitarie**: modelli di incentivazione basati su dati biometrici real-time, bonus per parametri cardiaci ottimali.

5. Ricerca clinica e trial

- **Studio di nuovi farmaci**: raccolta centralizzata di grandi volumi di ECG standardizzati, analisi "cloud-native" con ML per identificare pattern cardiotossici.
- **Registry aritmici**: costruzione di database multicentrici con ingestione serverless di ECG per training di modelli predittivi avanzati.

6. Home-care avanzato

- **Dispositivi patch-style** che raccolgono automaticamente 12 derivazioni per 24–48 h, inviano chunk ogni 10 s e attivano un allarme immediato se rilevano tachicardie o blocchi gravi.
- **Assistenza domiciliare per anziani**: integrazione con sistemi domotici e caregiver, per interventi rapidi in caso di cadute o svenimenti correlati a eventi cardiaci

Perché "serverless"?

- **Scalabilità**: passaggio da 10 a 10 000 pazienti senza riprogettare l'infrastruttura.
- **Costi ridotti**: si paga solo per l'inferenza e l'elaborazione effettivamente svolta.
- **Deployment rapido**: modelli aggiornabili on-the-fly con CI/CD (SageMaker, Cloud Run).