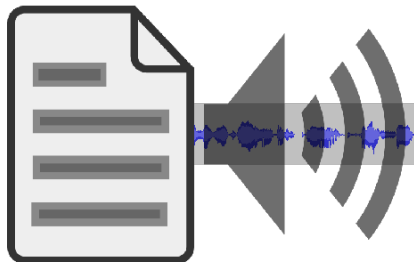




<https://www.mivoq.it/>

Giulio Paci <giulio.paci@mivoq.it>

Text To Speech

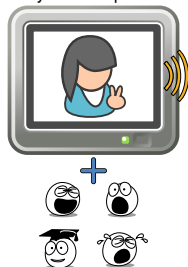


Crea la tua voce

Real Speech



Personalized
Synthetic Speech



Oggetto

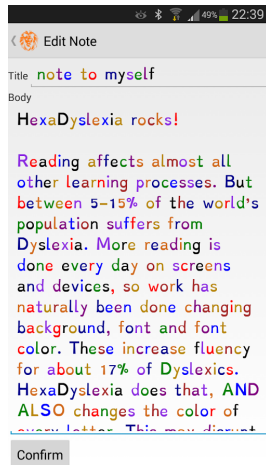
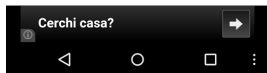


Problema

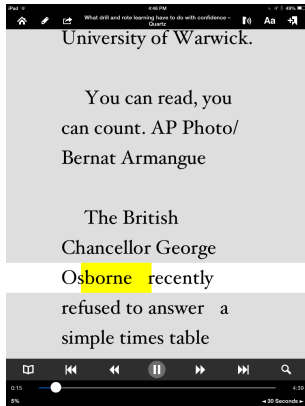
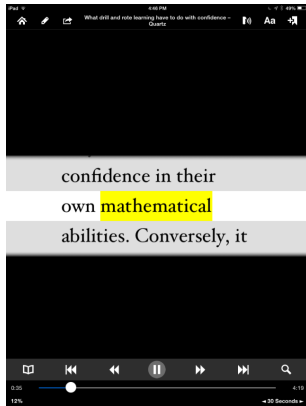
Uno studente per classe offre
di disturbi specifici
dell'apprendimento.
Vogliamo far e
qualcosa in loro.



Soluzione



Soluzione



Obiettivi di Mivoq

- ▶ **App utile** che utilizzi **Sintesi Vocale**
- ▶ Utilizzare parte del **codice** → opensource
- ▶ **Interfaccia Multimodale** (combinare audio e testo)

Sfida 1: Realizzare App funzionante

Integrare interfaccia **multimodale** interattiva

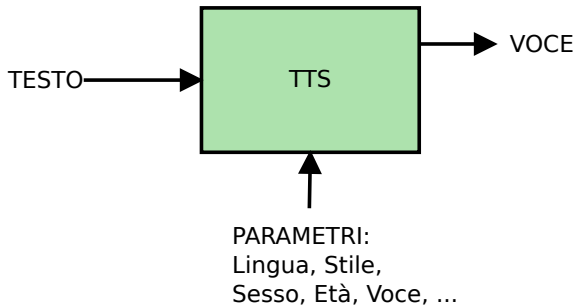


Lettoce di audiolibri
(es.: lettore di ePub)

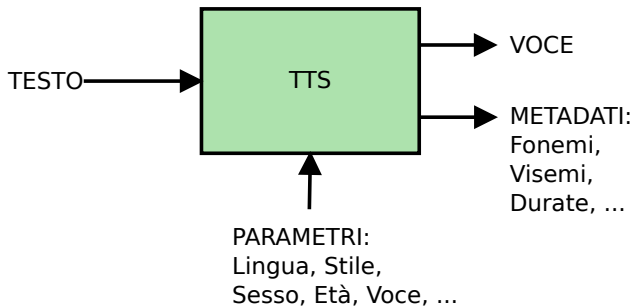


Applicazione di messaggistica
(es.: client Telegram)

Text To Speech



Text To Speech





<http://mediafi.org/>

FA-TTS: Flexible and Addaptive Text To Speech

- ▶ TTS fornito come servizio **SaaS**
- ▶ con supporto per: Inglese, Italiano, Tedesco, Francese, ...
- ▶ basato sul server **opensource** MaryTTS: bug fixes, miglioramenti vari, moduli personalizzati e voci personalizzate, nuove API **HTTP**
- ▶ **Flexible**: permette di cambiare lo stile della voce
- ▶ **Adaptive**: permette di replicare una voce specifica

Istruzioni per Installazione e Uso

- ▶ Le informazioni ufficiali sono disponibili su <http://mediafi.org/>

- ▶ **Per i più impazienti...**

<https://registry.hub.docker.com/u/fic2/fatts/>

- ▶ **Launch FA-TTS**

```
docker run -d -p 59125:59125 fic2/fatts
```

- ▶ Andate su <http://localhost:59125/> **e buon divertimento!**

API: get capabilities information: Version

```
curl --get http://localhost:59125/info/version
```

```
{
  "fa_tts_api_version": "0.0.1",
  "vendor": "Mivoq SRL",
  "product": "FA-TTS (MaryTTS server)",
  "specification": "5.2-SNAPSHOT",
  "implementation":
  {
    "revision": "5beaed"
  }
}
```

API: get capabilities information: Available languages

```
curl --get http://localhost:59125/info/locales/all
```

```
{
  "locales":
  [
    "de",
    "en",
    "en_US",
    "fr",
    "it",
    ...
  ],
  next: ...
}
```

API: get capabilities information: Available voices 1

```
curl --get http://localhost:59125/info/voices/all
```

```
{
  "voices":
  [
    {
      "id": "upmc-pierre-hsmm",
      "locales": ["fr"],
      "gender": "male"
    },
    ...
  ],
  "next": ...
}
```

API: get capabilities information: Available voices 2

```
curl --get http://localhost:59125/info/voices/locale/it
```

```
{
  "voices":
  [
    {
      "id":"istc-speaker_internazione-hsmm",
      "locales":["it"],
      "gender":"male"
    },
    ...
  ],
  "next": ...
}
```

API: get capabilities information: Voice capabilities

```
curl --get http://localhost:59125/info/voice/cmu-slt-  
hsmm/inputs/all
```

```
{  
  "inputs":  
  [  
    {"id":"TEXT"},  
    ...  
  ]  
}
```

API: get capabilities information: Voice capabilities

```
curl --get http://localhost:59125/info/voice/cmu-slt-hsmm/outputs/all
```

```
{
  "outputs":
  [
    {
      "id": "AUDIO",
      "formats":
      [
        {"id": "WAVE_FILE"},
        {"id": "AU_FILE"},
        {"id": "AU_STREAM"},
        {"id": "AIFF_FILE"}
      ]
    }
  ]
}
```

API: get capabilities information: Voice capabilities

```
curl --get http://localhost:59125/info/voice/cmu-slt-  
hsmm/styles/all
```

```
{  
  "styles":  
  [  
    {"id":"sad"},  
    {"id":"happy"},  
    ...  
  ]  
}
```


API: get capabilities information: Voice capabilities

```
curl --get http://localhost:59125/info/voice/cmu-slt-  
hsmm/effects/all
```

```
{  
  "effects": []  
}
```

API: get speech!

```
curl --get http://localhost:59125/say \
--data      input[type]=TEXT \
--data      input[locale]=it \
--data-urlencode input[content]="Hello_world." \
--data      output[type]=AUDIO \
--data      output[format]=WAVE_FILE \
--data      voice[gender]="neutral" \
--data-urlencode voice[name]="istc- \
                speaker_internazione-hsmm" \
--data      utterance[style]="sad" \
| aplay
```