A Deep Learning Based Tool For Ear Training

Project Management (GEP)
Assignment 4: Final Document

David Nogales Pérez

Bachelor Thesis
Specialisation in Computing

Director: Enrique Romero Merino
GEP Tutor: Paola Lorenza Pinto

March, 2022

Contents

1 Context and Scope **5**
- 1.1 Introduction . . . 5
- 1.2 Context . . . 5
- 1.3 Concepts . . . 5
  - 1.3.1 Deep Learning . . . 6
- 1.4 Problem to be Solved . . . 8
- 1.5 Stakeholders . . . 9
- 1.6 Justification . . . 10
  - 1.6.1 Current Solutions . . . 10
  - 1.6.2 Related Studies . . . 11
  - 1.6.3 Solution Justification . . . 12
- 1.7 Scope . . . 12
  - 1.7.1 Main Objective . . . 12
  - 1.7.2 Secondary Objectives . . . 12
  - 1.7.3 Additional Requirements . . . 13
  - 1.7.4 Risks and Obstacles . . . 13
- 1.8 Methodology and Rigour . . . 14
  - 1.8.1 Work Methodology . . . 14
  - 1.8.2 Monitoring Tools . . . 14

2 Project Planning **15**
- 2.1 Task Definition . . . 15

2.1.1 Resources 16
2.1.2 Project Management Tasks 17
2.1.3 General Tasks 18
2.2 Time Estimate 19
2.2.1 Workload Estimation 20
2.2.2 Gantt Diagram 22
2.3 Risk Management 23

3 Budget **24**
3.1 Cost Identification 24
3.1.1 Staff Costs 24
3.1.2 General Costs 25
3.1.3 Contingencies and Incidentals 26
3.2 Cost Estimates 26
3.3 Management Control 27

4 Sustainability Report **28**
4.1 Environmental Dimension 28
4.2 Economic Dimension 29
4.3 Social Dimension 29

List of Figures

1.1 Common Neural Network Structure. [3] 6
1.2 Example of Sequence-to-sequence tasks. [5] 7
1.3 RNNs internal architecture. [8] 8
1.4 Transformer model architecture. [10] 9
1.5 Melodic dictation User interface taken from teoria.com. [15] . . . 11

2.1 Task dependency graph from table 2.3 with resource colour coded (**PM**,green;**DS**,red and **FSD**,blue). [Own Compilation] 21
2.2 Estimated Gantt Diagram [Own Compilation] 22

List of Tables

1.1 Average negative log-probabilities in music datasets. [9] 8

2.1 Project events with their deadlines. [Own Compilation] 16
2.2 Time allocation by weeks and days. [Own Compilation] 20
2.3 Task dependencies and resource allocation. [Own compilation] . . 20

3.1 Salaries by role with 35% social security percentage.[19] 24
3.2 Human cost per activity (CPA). [Own Compilation] 25
3.3 Laptop amortization calculation. [Own Compilation] 25
3.4 General Costs Summary. [Own Compilation] 26
3.5 Incidental Cost Calculation. [Own Compilation] 26
3.6 Project Total Budget. [Own Compilation] 26

CHAPTER 1

Context and Scope

1.1 Introduction

In the following sections, the problem that this project aims to solve will be defined, as well as the theory necessary to understand it.

1.2 Context

The context in which this project is developed is as a Bachelor's thesis of the Computer Engineering Degree specialising in Computer Science, which is imparted at the Facultat d'Informàtica de Barcelona at the Universitat Politècnica de Catalunya. The project is overseen and mentored by Enrique Romero Merino, associate professor at the Department of Computer Science.

1.3 Concepts

Fundamental concepts regarding the scope of the project are defined in the following sections.

1.3.1 Deep Learning

Deep Learning, as defined in [1], is a subset of Machine Learning that provides different kinds of techniques and algorithms that allows computers to “learn” from great amounts of data. Neural Networks are the main algorithms of Deep Learning whose structure of layers of nodes is inspired by the human brain and its neurons [2]. An image of a common structure of a neural network is provided in Figure 1.1

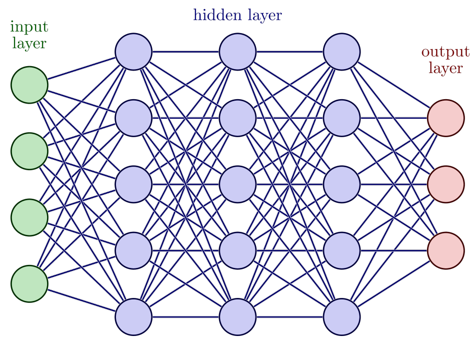

Figure 1.1: Common Neural Network Structure. [3]

Natural Language Processing (NLP)

Natural Language Processing refers to the study of natural language used daily by humans such as text or speech[4] and with the rise of computers and Deep Learning techniques, the study of this field has been greatly broaden.

Common problems tackled inside this field are:

- Text Classification.
- Language Modeling.
- Machine Translation.
- Speech Recognition.

Sequence-to-sequence Models

From an **NLP** perspective, Machine translation is a task mainly used to translate between a source and a target language used by different kinds of people. A **sequence-to-sequence** model is a more generalised approach to translation that, not only includes machine translation but other common tasks such as speech recognition, response generation, and practically anything [5]. A visual example of said tasks can be seen in Figure 1.2

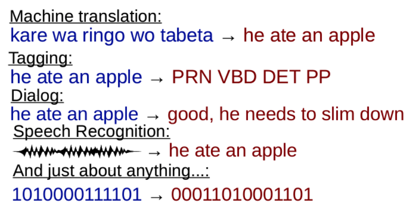

Figure 1.2: Example of Sequence-to-sequence tasks. [5]

Recurrent Neural Networks (RNNs)

RNNs are neural network architectures that differ from the common feedforward neural networks in mainly one aspect, they are oriented to time series or sequential tasks since its internal structure is suited to handle data with relationships through time.

Throughout the years, variations of the same architecture were introduced to tackle different issues that the original one presented, such as the **vanishing gradients** problem[1] which was addressed with the introduction of the **Long Short-Term Memory (LSTM)**[6] or the **Gated Recurrent Unit (GRU)**[7] which tried to maintain the gains of **LSTMs** while presenting a simplified architecture. A simple representation of each architecture can be seen in Figure 1.3.

An empirical comparison of said models was performed in [9], where a prediction task applied to music datasets was used as a benchmark between the models. The findings concluded in that the latter models performed equally well and outperformed the original **RNN** architecture. Some of the results can be seen in the table 1.1 where **GRU** model slightly outperforms the rest of the models.

[1] The vanishing gradients problem arises from the decaying error between the layers when performing backpropagation[6]

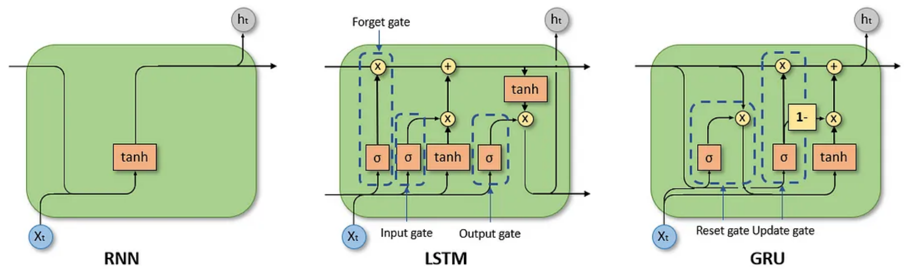

Figure 1.3: RNNs internal architecture. [8]

Music Dataset		tanh	GRU	LSTM
Nottingham	train	3.22	2.79	3.08
	test	**3.13**	3.23	3.20
JSBChorales	train	8.82	6.94	8.15
	test	9.10	**8.54**	8.67
MuseData	train	5.64	4.93	6.49
	test	6.23	**5.99**	6.23
Piano-midi	train	5.64	4.93	6.49
	test	9.03	**8.82**	9.03

Table 1.1: Average negative log-probabilities in music datasets. [9]

Transformers

With the introduction of the transformer architecture[10] (seen in Figure 1.4), models such as **BERT** (Bi-directional Encoder Representations from Transformers)[11] and **GPT** (Generative Pre-Training)[12], achieved **state-of-the-art** results on a wide range of **NLP**-related tasks.

Music Transformer[13] and **MusicLM**[14] are some examples of what can be achieved with this architecture when tackling problems in the music domain such as generating music with long-term structure or generating music from text descriptions.

1.4 Problem to be Solved

As a student at various music schools, I noticed an extensive lack of on-demand exercises that aim to develop one or more necessary basic skills to achieve a certain level of musicianship.

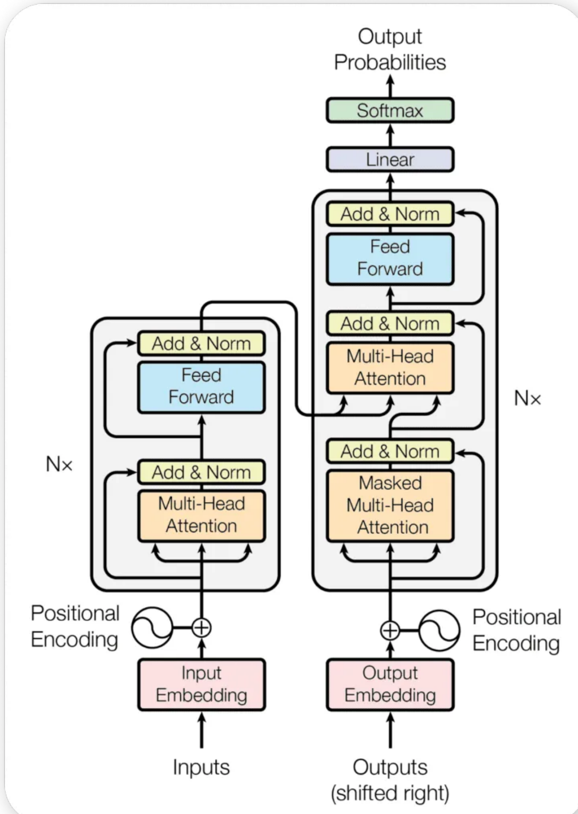

Figure 1.4: Transformer model architecture. [10]

This thesis intends to solve this particular issue by applying Deep Learning techniques to generate meaningful exercises for music teachers and their students. Since providing a technological solution for every type of exercise would not be feasible in the period of this thesis, the project will only focus on providing a solution for Melodic Dictation exercises.

1.5 Stakeholders

The different types of stakeholders involved in this project are listed below.

Primary

Music teachers and **students** are the main stakeholders in the project, as they are whom the outcome of this initiative is aimed at, providing them with tools that improve their workflows, in the case of teachers, or hone their skills, in the case of students.

Secondary

The author of the thesis and **the tutor** are both involved in the development of the project and have a shared interest in completing it successfully.

Tertiary

Music schools and **Conservatoires** could benefit from the outcome of the project but are not directly involved in it. Lastly, the **music community** since it could make use of the research poured into this project.

1.6 Justification

1.6.1 Current Solutions

Throughout my own journey of studying music composition, one of the most recommended websites by music teachers for practising ear training exercises has been **teoria.com**.[2]

The platform hosts different kinds of exercises such as melodic dictation or interval recognition. Looking at the user interface (Fig. 1.5) of one of the aforementioned exercises, one can tell that it allows the user to customise its practice session using a set of high level options.

When selecting different kinds of options a counter is provided to let the user know the number of exercises available that match the selected characteristics. The problem arises when the user selects very few options, leaving it with a small sample of exercises. In other words, the user will always have a **fixed** number of exercises when selecting some desired characteristics.

[2] **teoria.com** is a platform that specialises in providing music theory as well as a diverse range of practical exercises.

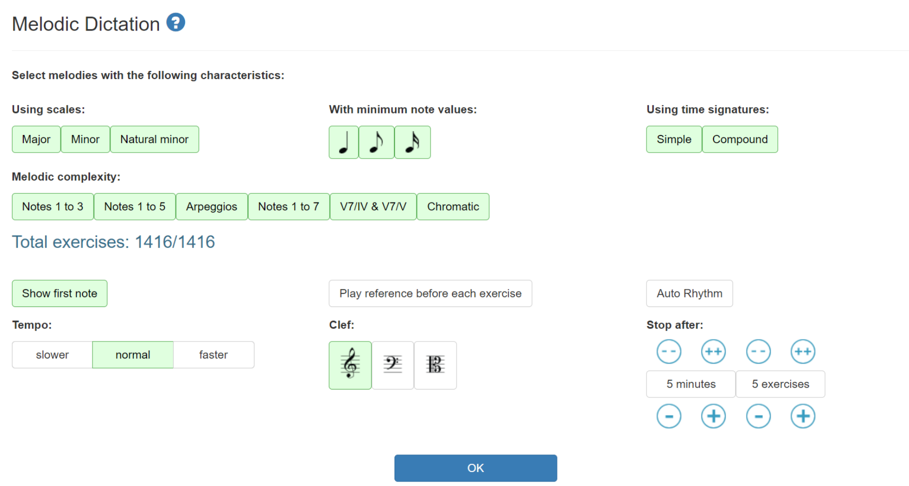

Figure 1.5: Melodic dictation User interface taken from teoria.com. [15]

In fact, the maximum amount of melodic dictation exercises that a user can practice, at least at the time of writing this thesis, are 1416 (See *Total exercises* in figure 1.5), which sounds like a decent amount, but when used regularly in sessions aimed specifically to practice a set of characteristics, it shows its lack of variety.

All in all, **teoria.com** is the better option when it comes to this particular type of exercise. Since it is a free alternative to traditional textbooks full of exercises and alternatives such as see ToneDEar whose exercises lack a much needed musicality[3].

1.6.2 Related Studies

In [16] we observed that **GPT-2** can be used to generate **believable music** when working with melodies represented in a text format such as the **ABC Notation**[4].

Additionally it has been demonstrated that **GPT-2** can be fine-tuned to generate music following a **specific structure** dictated by the user by using control codes [17]. This would mean that the model can be steered to create

[3] **Musicality** in this context meaning the quality of having a pleasant sound or melodiousness.
[4] **ABC Notation** is a system designed to notate music in plain text format.

melodies with a particular context such as the generation of melodic dictation exercises.

1.6.3 Solution Justification

As seen previously, the current solutions fail to provide **on-demand** melodic dictation exercises with sufficient variety and musicality when focusing in particular sets of characteristics.

Moreover, in the previous section we saw some studies which proved that **GPT-2** can generate interesting melodies in a controlled manner. This would mean that we could leverage the power of language models to potentially solve our problem.

1.7 Scope

1.7.1 Main Objective

As previously introduced in Section 1.4, the main objective of this thesis is to present a tool capable of generating melodic dictation exercises with an specific structure and sufficient musicality.

1.7.2 Secondary Objectives

To accomplish the main objective, the development of the project has been partitioned into secondary objectives:

- Research transformer based Language Models.
  - Study transformer architecture.
  - Explore different frameworks for manipulating said models.
- Familiarise with **NLP** techniques.
- Train the model on a suitable dataset.
  - Study different music datasets.
  - Fine-tune the model to the dataset
- Deploy the model to a web application.
  - Research web development stacks.

  - Develop user interface.
  - Program API server.

- Empirically evaluate obtained results.

1.7.3 Additional Requirements

Functional Requirements

- The model is capable of generating melodic dictation exercises.
- The website is capable of interacting with the model
- The website provides a music sheet viewer for the generated exercise.

Non Functional Requirements

- The exercises have sufficient musicality.
- The exercises follow the characteristics provided by the user
- The exercises are generated in a timely manner.
- The application is user-friendly.
- The app has a responsive web design.
- The project follows an Agile methodology and good programming practices.

1.7.4 Risks and Obstacles

Throughout the development of the project some problems may arise that have to be taken into account. Potential issues are described as follows:

- **Project deadline.** For each phase of the project there are different deadlines to be met.
- **Inexperience in the domain.** Given that the contents of the project are not studied deeply throughout the degree, the learning curve of the subject has to be considered since more time than the allocated could be needed.

- **Poor planning.** Less time than needed could be assigned to one or more tasks, causing delays in the project.
- **Computational power.** Deep learning models often require a considerable amount of resources that may not be readily available.

1.8 Methodology and Rigour

1.8.1 Work Methodology

For the development of this project I chose to follow an agile methodology since it provides a more flexible environment compared to a waterfall approach.

More specifically the project will use the Scrum framework [18], with *sprints* of one week and with all the roles assigned to myself. At the end of each week the common Scrum events (*Sprint Planning, Review and Retrospective*) will take place. The Daily Scrum will not take place since all the roles are performed by one person.

1.8.2 Monitoring Tools

To monitor the project, we will use **Git** as the main version control software, **Overleaf** to keep track of the report and **GitHub** as the platform for remotely hosting code and as a redundancy measure. Additionally, **GitHub Projects** will be used to keep track of the project following the Scrum Methodology.

Finally, meetings with the tutor will always be scheduled before reaching a milestone, such as passing the project management course, the control meeting stipulated in the project regulations, and before the final presentation. But they will not be regularly scheduled, in an attempt to emulate real-life conditions of the development of a project, aiming to encourage work independence and creativity in finding solutions when facing complex problems.

CHAPTER 2

Project Planning

2.1 Task Definition

The Bachelor thesis of the Computer Science degree (**TFG**) at **UPC** has a work load of 18 **ECTS**, 3 of which belong to the project management course, which according to its syllabus, it is estimated to be 75 working hours.

This would mean that each **ECTS** is worth 25 hours of work, so the overall project should comprise at least 450 working hours, of which 375 hours are allocated to project development and the rest to the project management course.

The course syllabus also proposes to allocate 37.5 hours of its workload to the study of the course material and the synthesis of the final course report, while the remaining 37.5 hours should be dedicated to working on the main tasks of the **TFG**.

Additionally, the project has to meet different deadlines throughout its development. First, the student has to pass the Project Management course, then a follow-up meeting with the tutor has to take place to be allowed to pick a presentation date.

Furthermore, due to personal reasons, the development of this project had to be delayed for a couple of months and therefore the deadlines of a normal term had to be changed accordingly. The deadlines for each phase of the project can be at table 2.1.

Event	Deadline
Project Management Report Delivery	15/03/2023
Progress Review	14/04/2023
Memory Presentation	08/05/23-12/05/2023
Project Defence	15/05/23-19/05/2023

Table 2.1: Project events with their deadlines. [Own Compilation]

In the following sub-sections the tasks will be partitioned according to their nature and assigned an estimation in hours. Moreover, dependencies between tasks will be described as well as human and material resources requirements.

2.1.1 Resources

Material Resources

A wide range of tools will be used throughout the development the project as described down below.

- **Code Editor**. VS Code will be used as the main tool for writing and editing code, given its ease of use and extensibility.
- **Report Editor.** Overleaf will be used since it provides an intuitive interface for writing *LaTex* based documents.
- **Version Control Software.** Git and GitHub will be used to keep track of changes in the project locally and remotely.
- **Project Monitoring Tools.** GitHub Projects will be used to supervise the project and Google Workspace apps will be used to arrange meetings and share progress with the tutor.
- **Development Platform.** Docker will be used to containerise the environment of each aspect of the project.

- **Hardware.** A laptop will be used with the following specifications: 16 GB RAM, Intel Core i7-10870H and NVIDIA GeForce RTX 3060.

Human Resources

During the development of the project different human resources will intervene. Said resources are describe as follows:

- **Project Manager (PM).** In charge of planning the project and ensuring that the objectives are being met.
- **Data Scientist (DS).** Designs and develops the necessary components for an artificial intelligence project. Some responsibilities of a data scientist are gathering data, building pipelines to use said data, and the final model evaluation.
- **Full Stack Developer (FSD).** Is in charge of developing both client and server software for a website.
- **AI Consultant (AIC).** Provides expert knowledge in the domain. This role will be performed by the tutor

In table 2.3, one can observe the different roles assigned to its corresponding task.

2.1.2 Project Management Tasks

The tasks related to the overall planning of the project are described below:

- **Tutor Meetings.** The meetings will only take place to present meaningful progress in the project and occasionally to solve doubts.
- **Contextualisation and Scope.** Defines the objectives of the project, its relevance, and the context of the study.
- **Project Planning.** Describes the tasks to be solved throughout the project and all the necessary resources for its completion
- **Economic Management.** Analyses the economic cost of undertaking a project of this nature.
- **Sustainability Report.** Analyses the sustainability of the project given the resources poured into it.

- **Final Document Synthesis.** Verifies the redaction quality of the final report to make sure it gathers all the requirements of a **TFG**.

- **Development Monitoring.** Comprises all the events of the Scrum framework necessary to correctly monitor the development of the project.

2.1.3 General Tasks

The objectives described in section 1.7.2 are described down below.

Research

- **Study transformer Architecture.** Familiarise myself with the theory behind the architecture and some popular transformer-based language models.

- **Familiarise with NLP techniques.** Study different techniques on how to handle text will help to treat the data in a more efficient way.

- **Analyse Music Datasets.** Select a suitable dataset for the project since a proper dataset should avoid doing more work than necessary.

- **Explore Deep Learning Frameworks.** Pick an appropriate framework is an important task given that it could greatly reduce the development time since they are aimed to provide different layers of abstraction to solve a variety of problems.

- **Investigate web development stacks.** Evaluate the strengths and weaknesses of different libraries when deploying a Deep Learning application.

Development and Experimentation

- **Create Development Environments.** The necessary software for the project has to be defined and installed in order to proceed with the rest of tasks.

- **Exploratory Data Analysis (EDA).** This task is necessary to gain valuable insights on the selected dataset and to pre-process it correctly.

- **Build Experimentation Pipeline.** Common Machine Learning Pipelines involve the development of functions with different objectives, such as a data loader, data pre-processor, model trainer, model fine-tuner and a model evaluator. All of these are necessary for creating a model, so it is essential to build a robust pipeline.

- **Train Model.** This process consumes a considerable amount of time and utilises the experimentation pipeline.
- **Fine-tune Model.** Once trained, the model is evaluated and fine-tuned to generate the best possible results.
- **Design User Interface.** A sketch of the interface is necessary to proceed with its implementation.
- **Implement User Interface.** The front-end is implemented following a concrete design.
- **Program API Server.** The back-end is implemented to allow interaction with the trained model.
- **Deploy web application.** The full application is deployed using a suitable platform.

Testing and Evaluation

- **Evaluate Model.** Empirically evaluate the obtained results and draw a conclusion
- **Test User Interface Design.** Assess interface usability.

2.2 Time Estimate

As said previously in Section 2.1, different workloads and deadlines have to be taken into account in order to develop the project successfully.

In order to have balanced workloads every week, the amount of work will be evenly distributed between weeks starting at **02/01/2023** until the memory presentation (See 2.1) which roughly comprises 17 weeks.

The work in hours per week and day is shown in table 2.2, where it is estimated that a typical day will need **4 hours** of work to reach (and surpass) the 450 hours stipulated by the university normative.

It is worth noticing that the rounded estimation is an approximation of the project objective and that the hours allocated for the project management course are there to reflect that some weeks may need approximately **5 hours** to be dedicated in order to do both at the same time.

	Total (h)	Working Weeks	Hours per Week (h)	Working Days	Hours per Day (h)
Project Objective	412.5	17	24.3	7	3.5
Project Management Course	37.5	4	9.4	7	1.3
Rounded Estimation	476	17	28	7	4

Table 2.2: Time allocation by weeks and days. [Own Compilation]

2.2.1 Workload Estimation

The estimated workload for each task is provided in table 2.3. Additionally a dependency graph is provided (See Figure 2.1) to assess the concurrency of the tasks efficiently.

It is worth noting that tasks T1.1 and T1.7 are missing from the aforementioned graph given that they can be performed in every step of the development.

Name	Task	Time (h)	Dependencies	Human Resources
Project Management	**T1**	**136**		
Tutor Meetings	T1.1	10	-	PM, AIC
Contextualisation and Scope	T1.2	15	-	PM, DS
Project Planning	T1.3	12	T1.2	PM
Economic Management	T1.4	12	T1.3	PM
Sustainability Report	T1.5	12	T1.4	PM
Final Document Synthesis	T1.6	40	T1.5, T4.1, T4.2	PM
Development Monitoring	T1.7	35	-	PM
Research	**T2**	**200**		
Study transformer Architecture	T2.1	60	T1.2	DS
Familiarise with NLP techniques	T2.2	50	T1.2	DS
Analyse Music Datasets	T2.3	50	T1.2	DS
Explore Deep Learning Frameworks	T2.4	20	T2.1, T2.2	DS
Investigate web development stacks	T2.5	20	-	FSD
Development and Experimentation	**T3**	**100**		
Create Development Environments	T3.1	5	T2.4, T2.5	DS
Exploratory Data Analysis (EDA)	T3.2	10	T2.3, T3.1	DS
Build Experimentation Pipeline	T3.3	20	T3.2	DS
Train Model	T3.4	25	T3.3	DS
Fine-tune Model	T3.5	10	T3.4	DS
Design User Interface	T3.6	10	-	FSD
Implement User Interface	T3.7	10	T3.6, T3.1	FSD
Program API Server	T3.8	5	T3.1	FSD
Deploy web application	T3.9	5	T3.8, T3.7	FSD
Testing and Evaluation	**T4**	**40**		
Evaluate Model	T4.1	20	T3.4, T3.5	DS, AIC
Test User Interface Design	T4.2	20	T3.9	FSD

Table 2.3: Task dependencies and resource allocation. [Own compilation]

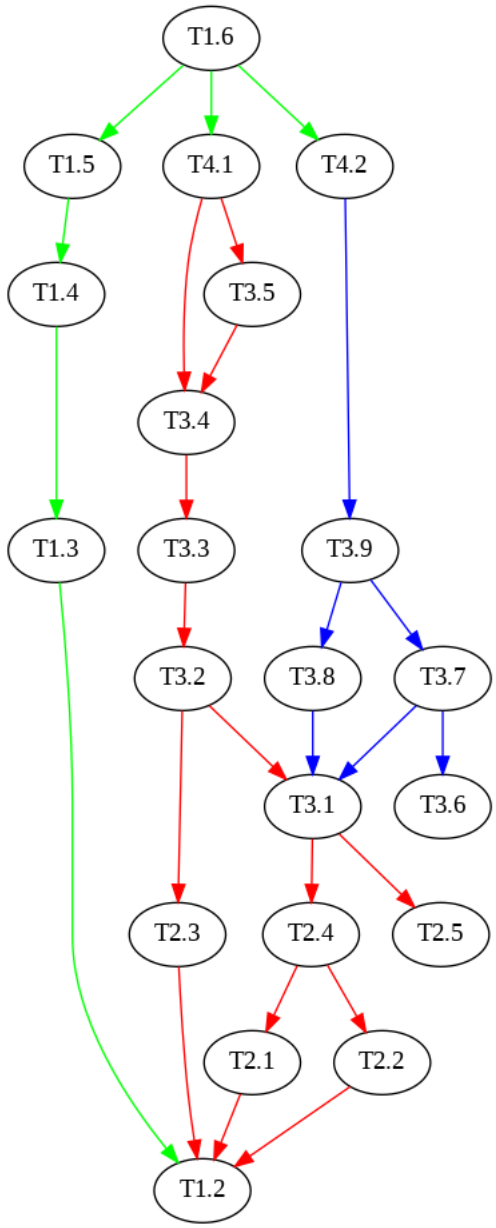

Figure 2.1: Task dependency graph from table 2.3 with resource colour coded (**PM**,green;**DS**,red and **FSD**,blue). [Own Compilation]

2.2.2 Gantt Diagram

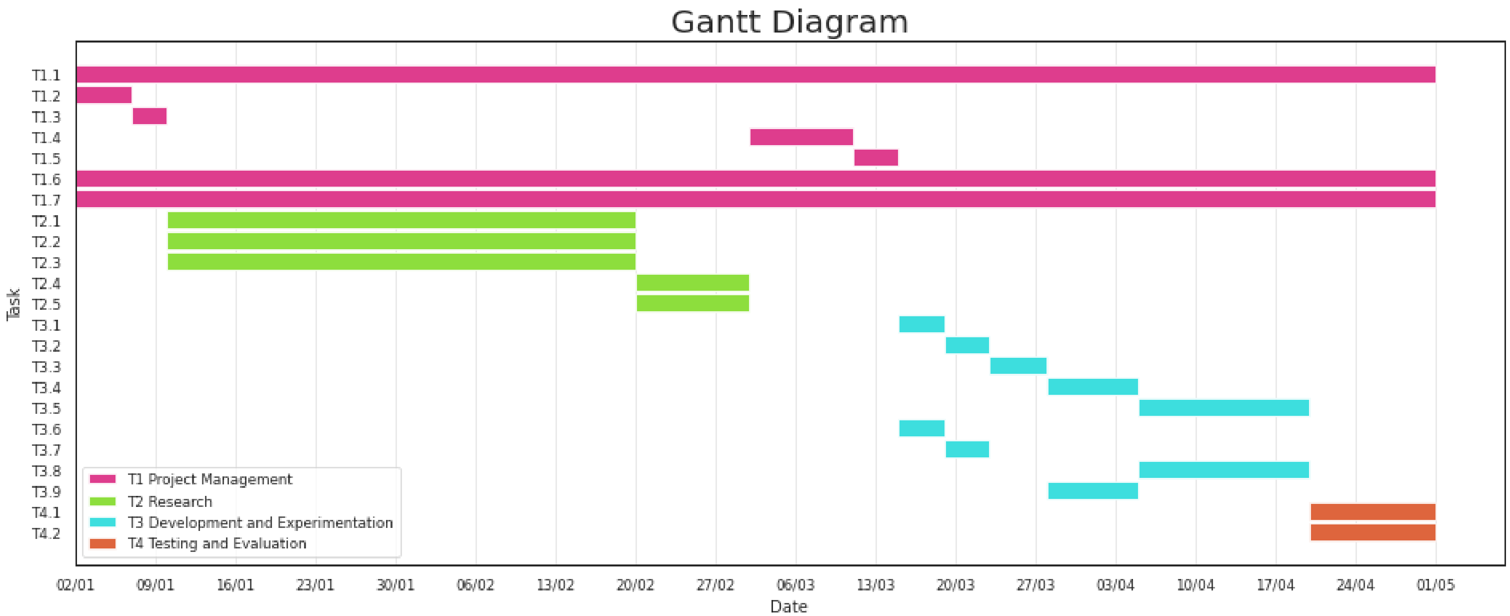

Figure 2.2: Estimated Gantt Diagram [Own Compilation]

2.3 Risk Management

As previously discussed in Section 1.7.4, one has to contemplate possible ways to solve such problems. An assessment of the impact of each problem is provided below, accompanied by a recommended solution.

- **Project deadline [Low Impact].** If a deadline is approaching, it is possible to increase the number of hours per day for any task, as all planning has been calculated with this option in mind. Furthermore, this issue is constantly evaluated at the end of each sprint to re-evaluate the workloads.
- **Inexperience in the domain [High Impact].** More hours can be allocated to research if necessary, but this would mean that some non-functional requirements would not be developed or even that the quality of the project itself would be lowered, resulting in a reduction of the project scope.
- **Poor planning [Low Impact].** By following an Agile Methodology this problem is greatly mitigated as the workload of each task is reviewed in a timely manner to avoid delays.
- **Computational power [Medium Impact].** If a deadline is approaching, one can evaluate the possibility of moving the local training to a paid cloud-based environment to accelerate the model training. But in addition to this being expensive in some cases, we would also have to adapt the code to run on the remote instance.

CHAPTER 3

Budget

3.1 Cost Identification

This section analyses the budget necessary to carry out the project in its entirety.

3.1.1 Staff Costs

In section 2.1.1, we have already identified the different roles in the project and also assigned them their corresponding tasks (See 2.3).

The different salaries for the roles are shown at the table 3.1. The annual salary and hourly fees are described with and without taking into account the social security percentage. It is worth noting that the hourly rate was calculated considering a 40 hours of work each week and 52 weeks in a year.

Role	Annual Salary (€)	Annual Salary with Social Security (€)	Hourly Rate (€/h)
Project Manager (PM)	48452	65410.20	31.45
Data Scientist (DS)	35666	48149.10	23.15
Full Stack Developer (FSD)	22481	30349.35	14.59
AI Consultant (AIC)	53117	71707.95	34.47

Table 3.1: Salaries by role with 35% social security percentage.[19]

In table 3.2 we can observe the cost per activity (**CPA**) based on the hourly fees of each role while taking into account the social security payments.

Name	Task	Time (h)	PM (h)	DS (h)	FSD (h)	AIC (h)	Cost Per Task (€)
Project Management	**T1**	**136**					
Tutor Meetings	T1.1	10	10	0	0	10	659.22
Contextualisation and Scope	T1.2	15	15	15	0	0	818.94
Project Planning	T1.3	12	12	0	0	0	377.37
Economic Management	T1.4	12	12	0	0	0	377.37
Sustainability Report	T1.5	12	12	0	0	0	377.37
Final Document Synthesis	T1.6	40	40	0	0	0	1257.89
Development Monitoring	T1.7	35	35	0	0	0	1100.65
Research	**T2**	**200**					
Study transformer Architecture	T2.1	60	0	60	0	0	1388.92
Familiarise with NLP techniques	T2.2	50	0	50	0	0	1157.43
Analyse Music Datasets	T2.3	50	0	50	0	0	1157.43
Explore Deep Learning Frameworks	T2.4	20	0	20	0	0	462.97
Investigate web development stacks	T2.5	20	0	0	20	0	291.82
Development and Experimentation	**T3**	**100**					
Create Development Environments	T3.1	5	0	5	0	0	115.74
Exploratory Data Analysis (EDA)	T3.2	10	0	10	0	0	231.49
Build Experimentation Pipeline	T3.3	20	0	20	0	0	462.97
Train Model	T3.4	25	0	25	0	0	578.72
Fine-tune Model	T3.5	10	0	10	0	0	231.49
Design User Interface	T3.6	10	0	0	10	0	145.91
Implement User Interface	T3.7	10	0	0	10	0	145.91
Program API Server	T3.8	5	0	0	5	0	72.96
Deploy web application	T3.9	5	0	0	5	0	72.96
Testing and Evaluation	**T4**	**40**					
Evaluate Model	T4.1	20	0	20	0	20	1152.47
Test User Interface Design	T4.2	20	0	0	20	0	291.82
		Hours per Role (h)	136	285	70	30	Total Cost (€)
		Cost per Role (€)	4276.82	6597.35	1021.37	1034.25	**12929.80**

Table 3.2: Human cost per activity (CPA). [Own Compilation]

3.1.2 General Costs

In order to reflect the costs of working remotely, internet and electricity costs are calculated as follows:

- **Internet Cost.** Monthly invoice costs around 55€. Therefore, 5 months worth of internet will cost 275€
- **Electricity Cost.** The power consumption of a laptop is around 0.2 kWh. Electricity cost in Spain is around 0.228 €/h, given that the laptop will be used for 476 hours, the estimated electricity cost will be 21.71 €.

Additionally, the laptop depreciates by the passing of time and by its use so we have to take into account its amortisation. Table 3.3 shows the laptop amortisation.

Hardware	Initial Cost (€)	Life Expectancy (Years)	Yearly Usage (h)	Hours Used (h)	Amortisation (€)
Laptop	1200	5	2080	476	**54.92**

Table 3.3: Laptop amortization calculation. [Own Compilation]

One can see the General Costs (GC) calculated in table 3.4 for hardware and cost of electricity, software is not taken into account since all the software described in section 2.1.1 is free to use or at least, a free tier is available.

Type	Cost (€)
Hardware	
Laptop amortisation	275.00
Space	
Electricity	275.00
Internet	21.71
Total GC	**351.63**

Table 3.4: General Costs Summary. [Own Compilation]

3.1.3 Contingencies and Incidentals

In order to lessen the impact of unexpected events, a 10% contingency will be applied to the CPA, as it is common practice to do so in project of this nature.

Additionally, incidentals like those seen in section 1.7.4 can occur. For this, the risk of this events has to be calculated and also has to be provided with part of the budget. These calculations can be seen in table 3.5.

	Estimated Cost (€)	Risk (%)	Cost (€)
Project Deadline	1100.65	10	110.07
Inexperience in the Domain	1388.92	20	277.78
Poor Planning	1100.65	10	110.07
Computational Power	578.72	15	86.81
		Total Cost	**584.72**

Table 3.5: Incidental Cost Calculation. [Own Compilation]

3.2 Cost Estimates

An estimation for the entirety of the project is shown in table 3.6 as a summary of all the previous tables.

Activity	Cost (€)
Total CG	351.63
Total CPA	12929.80
CPA Contingency	1939.47
Total Incidentals	584.72
Total Budget	**15805.62**

Table 3.6: Project Total Budget. [Own Compilation]

3.3 Management Control

In order to control possible budget deviations, a set of formulas is presented with the aim of providing a way to detect possible imprecisions in our estimations.

- **CPA Deviation (CPAD).** $CPAD = (estimated\_cost\_per\_hour - real\_cost\_per\_hour) * total\_hours\_consumed$
- **CG Deviation (CGD).** $CGD = ED + AD$
- **Electricity Deviation (ED).** $ED = (estimated\_usage\_per\_hour - real\_usage\_per\_hour) * price\_per\_hour$
- **Amortisation Deviation (AD).** $AD = (estimated\_hours\_used - real\_hours\_used) * price\_per\_hour$
- **Contingency and Incidental Deviation (CID).** $CID = (estimated\_incidental\_hours - real\_incidental\_hours) * total\_incidental\_hours$
- **Cost Deviation (CD).** $CD = estimated\_cost - (CPAD + CGD + CID)$

CHAPTER 4

Sustainability Report

4.1 Environmental Dimension

Have you estimated the environmental impact of undertaking the project? Have you considered how to minimise the impact, for example by reusing resources?

No, only human and material resources have been estimated, as measuring the environmental impact of the project would have been a waste of time due to the lack of tools to do so with sufficient precision. But transfer learning has been contemplated as a way of reusing resources in the context of the project.

How is the problem that you wish to address resolved currently (state of the art)? In what ways will your solution environmentally improve existing solutions?

As previously seen in section 1.6.2 there are some related studies that reused **GPT-2** for their own tasks, so one way of reusing resources in this context would clearly be to apply **transfer learning** where possible.

My solution proposes to use these pre-trained language models to offer a better alternative to current solutions, while avoiding training a model from scratch, thus saving resources.

4.2 Economic Dimension

Have you estimated the cost of undertaking the project (human and material resources)

Yes, in section 2.1.1 the necessary resources are defined and in chapter 3 is devoted to estimate different kinds of costs for the project.

How is the problem that you wish to address resolved currently (state of the art)? In what ways will your solution economically improve existing solutions?

My solution would offer a free alternative to paid content, such as books full of exercises. It would also save time within the classroom, as the teacher would not have to waste time creating new exercises, and would also save the music student money, as studying music can be quite expensive, that time spent with the teacher should not be wasted on a task that could be automated.

4.3 Social Dimension

What do you think undertaking the project has contributed to you personally

As a music student myself and failing in exactly the area that this tool is intended to help, I believe that when it is finished it will not only help me, but other students as well. Furthermore, finding this intersection between music and technology has made me appreciate both fields even more.

How is the problem that you wish to address resolved currently (state of the art)? In what ways will your solution socially improve (quality of life) existing? Is there a real need for the project?

I think the project could potentially help other music students besides myself, as it is an issue I struggle with, I think at least someone else may have the same problem as me so yes, I think there is a real need for the project.

Bibliography

1. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* http://www.deeplearningbook.org (MIT Press, 2016).
2. Fitch, F. B. McCulloch Warren S. and Pitts Walter. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics, vol. 5, pp. 115–133 (1944).
3. Neutelings, I. *Neural networks with tiks* https://tikz.net/neural_networks/.
4. Bird, S., Klein, E. & Loper, E. *Natural Language Processing with Python* ISBN: 9780596516499. https://books.google.es/books?id=ScL3wAEACAAJ (O'Reilly Media, 2009).
5. Neubig, G. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619* (2017).
6. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9,** 1735–1780 (1997).
7. Cho, K., Van Merriënboer, B., Bahdanau, D. & Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).
8. Clay-Atlas. *Graphical Introduction Note About GRU* https://clay-atlas.com/us/blog/2021/07/27/gru-en-introduction-note/.

9. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

10. Vaswani, A. *et al.* Attention is all you need. *Advances in neural information processing systems* **30** (2017).

11. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

12. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., *et al.* Improving language understanding by generative pre-training (2018).

13. Huang, C.-Z. A. *et al.* Music Transformer: Generating Music with Long-Term Structure. *arXiv preprint arXiv:1809.04281* (2018).

14. Agostinelli, A. *et al.* MusicLM: Generating Music From Text. *arXiv preprint arXiv:2301.11325* (2023).

15. Alvira, J. R. *teoria.com is a website dedicated to the study and practice of music theory.* https://www.teoria.com/.

16. Geerlings, C. & Meroño-Peñuela, A. *Interacting with GPT-2 to Generate Controlled and Believable Musical Sequences in ABC Notation* in *NLP4MUSA* (2020).

17. Wu, S. & Sun, M. *TunesFormer: Forming Tunes with Control Codes* 2023. https://arxiv.org/abs/2301.02884.

18. Schwaber, K. & Sutherland, J. *The 2020 Scrum Guide* https://scrumguides.org/download.html.

19. PayScale. *Salary Comparison, Salary Survey, Search Wages* https://www.payscale.com/research/ES/.