

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Inven!RA – Platform for authoring and tracking of Inventive Activity Plans

Tiago José Lima Cruzeiro



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Leonel Morgado

Second Supervisor: António Coelho

July 22, 2020

Inven!RA – Platform for authoring and tracking of Inventive Activity Plans

Tiago José Lima Cruzeiro

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

External Examiner: Mário Fontes

Supervisor: Leonel Morgado

Second Supervisor: António Coelho

July 22, 2020

Resumo

A gestão e integração de tecnologia pervasiva e imersiva como parte de atividades educativas continua um problema por resolver no panorama atual de tecnologia na educação. Neste trabalho, expandimos sobre a abordagem do projeto internacional BEACONING, que aplicou uma ferramenta de autoria para permitir a “designers educativos” criar “Planos de Aula *Gamificados*”, combinando diversos minijogos produzidos por terceiros, para professores e alunos aprenderem enquanto jogam, com o progresso acompanhado por uma plataforma tecnológica.

Levamos este conceito mais longe, ao acompanhar diversas atividades, não apenas jogos, e chamamos-lhe Inven!RA, com o objetivo de suportar “Ensino Inventivo”. Focamo-nos numa plataforma aberta para várias atividades de terceiros, que são semelhantes no facto de fornecerem a líderes de atividades (por ex. professores) com o estado e progresso dos participantes (por ex. alunos) em objetivos de aprendizagem. O contributo é uma arquitetura para suportar ferramentas de autoria, atividades *plug-in*, e seguimento de analíticas para planos de atividades inventivas, desenvolvida e validada com um protótipo funcional, integrado com atividades externas, como um sistema aberto e expansível.

A Inven!RA é um novo conceito, ainda na sua infância, a ser desenvolvido sob a alçada de cooperação internacional de investigação, com aconselhamento de investigadores da Immersive Learning Research Network e do projeto PRINT/Transformação Digital e Humanidades da universidade brasileira UNISINOS. Estivemos sempre inspirados em contribuir significativamente para o mundo da aprendizagem imersiva, e esperamos ter sido bem-sucedidos no lançamento inicial deste conceito.

O trabalho aqui documentado foi grandemente enriquecido pela integração contínua com outros projetos e colegas, que ajudaram a validar a solução e a conclusão dos seus objetivos, bem como provar a flexibilidade do sistema proposto.

Abstract

Managing and integrating immersive and pervasive technology as part of educational activities remains an unsolved problem in today's educational technology panorama. In this work, we build upon the approach of the BEACONING project, which employed an authoring tool to enable learning designers to create Gamified Lesson Plans combining several third-party minigames, for teachers and students to learn while playing games, with progress tracked by a technological platform.

We take this concept further to support diversified activities, not just games, and call it Inven!RA, towards the ultimate goal of supporting Inventive Learning. Thus, we focus on an open platform for multiple third-party activities, with the common ground of providing activity leaders (e.g., teachers) with status tracking of participants (e.g., students) towards learning objectives. The contribution is an architecture to support authoring tools, plug-in activities, and analytics tracking of for inventive activity plans, developed and validated via a working prototype, integrated with third-party activities, as an open and expandable ecosystem.

Inven!RA is a new concept, still in its infancy, being developed under the wing of international research cooperation, with advice from researchers from the Immersive Learning Research Network and from the project PRINT/“Digital Transformation and Humanities” of the Brazilian UNISINOS university. We were inspired to give a significant contribution to the world of immersive learning and hopefully succeed in kickstarting the concept.

This work was greatly enriched and benefited by continuous integration with other projects and colleagues, which helped validate the solution and its set goals, as well as prove the flexibility of the system.

Acknowledgments

I need to start by thanking my parents, *José* and *Fátima*, for enabling me to study, always supporting me with patience and love, for trusting and respecting my choices and never giving up.

To my extended family, for helping me with all they could and inspiring me.

To my friends, the old ones, the ones I made along the way and the ones that left too soon, you enriched this journey and helped make each day brighter.

To my work colleagues at *eProseed*, for being understanding and supporting me when I had to share my focus.

To my supervisors, *Professor Leonel Morgado* and *Professor António Coelho*, for giving me this opportunity and guiding me through it with their knowledge, while being understanding.

To Filipa, the love of my life, for her unyielding support, care, patience and strength. You always made me push my limits and I wouldn't be here without you.

Tiago Cruzeiro

*“The roots of education are bitter,
but the fruit is sweet.”*

Aristotle

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation and Objectives	2
1.3	The Challenge	2
1.3.1	Overview	2
1.3.2	Research Questions	3
1.3.3	The importance of an open ecosystem	3
1.4	Structure	3
2	Background	5
2.1	Managing dynamic educational activities	5
2.2	Related Work	6
2.2.1	BEACONING	6
2.2.2	Others	8
2.3	Analytics harvesting	10
2.3.1	Approaching the certification of competences	11
2.4	Technologies overview	12
2.4.1	Backend	13
2.4.2	Frontend	15
2.5	Summary	17
3	Getting to the solution	19
3.1	How we raised requirements	19
3.2	User Stories	20
3.2.1	US01 - Learning designer creates a linear IAP	20
3.2.2	US02 - Learning designer creates a non-linear IAP: Escape Room	21
3.2.3	US03 - Learning designer edits previously saved IAP	22
3.2.4	US04 – Teacher deploys IAP to a class	22
3.2.5	US05 – Student experiences IAP	22
3.2.6	US06 – Teacher tracks the audience’s progress on a given IAP	22
3.3	Activity Diagrams	22
3.3.1	Creating and Editing Activity	23
3.3.2	Deploying Activity	24
3.3.3	Student experiences IAP Activity	25
3.3.4	Tracking IAP Analytics Activity	26
3.4	System Requirements	27

4	Proposed Solution	29
4.1	Concept	29
4.2	Useful terminology	30
4.3	High Level Description	31
4.4	System Architecture	32
4.4.1	Backend	34
4.4.2	Frontends	35
4.5	Sequence Diagrams	36
4.5.1	Insert activity in a plan	36
4.5.2	Assign pedagogic objective to a plan	37
4.5.3	Connect activities	38
4.5.4	IAP Deployment	39
4.5.5	Student Activity Deployment	40
5	The prototype	43
5.1	Software Development Life Cycle	43
5.2	Backend	44
5.2.1	Services	45
5.2.2	Running the backend	48
5.3	Frontend	49
5.3.1	Implementation	50
6	Validation and testing	55
6.1	Unit Testing	55
6.2	Visual testing	55
6.2.1	Navigation and Homepage	57
6.2.2	Deployed IAPs view	58
6.2.3	Registered activities view	59
6.2.4	Edit IAP view	60
6.2.5	IAP Analytics	63
6.3	Integration Testing	64
6.3.1	inventiveTr@ining by Duarte Cota	64
6.3.2	CHIC's apps by Rui Quaresma	66
6.4	Results Summary	69
7	Conclusions and future work	71
7.1	Achieved objectives	71
7.2	Limitations	72
7.3	Future Work	72
A	Setup Guide	75
A.1	Backend	75
A.1.1	Pre-Requisites	75
A.1.2	Installation steps	75
A.2	Frontend	76
A.2.1	Pre-Requisites	76
A.2.2	Installation steps	76
A.3	Serving the application using NGINX	77

B	REST Specification	79
B.1	REST API	79
B.1.1	IAP Services	79
B.1.2	Activity Services	81
B.1.3	Deployment Services	82
B.2	Sample Payloads	84
B.2.1	Sample Response List IAPs	84
B.2.2	Sample Response Get IAP	86
B.2.3	Sample Request Create IAP	88
B.2.4	Sample Request List Activities	89
B.2.5	Sample Request Create Activity	92
B.2.6	Sample Request Get Activity	92
B.2.7	Sample deploy IAP	93
B.2.8	Sample list activities for deployed IAP	95
B.2.9	Sample list activities for deployed IAP	95
B.3	JSON Schemas	96
B.3.1	IAP Schema	96
B.3.2	Activity	107
	References	113

List of Figures

2.1	GLP Data Model proposed in [34]	8
2.2	BEACONING's Authoring Tool, extracted from [1]	9
2.3	Adding an activity on BEACONING, extracted from [1]	10
2.4	Twine 2 Story Map [7]	11
2.5	Padlet Sample Narrative [12]	12
2.6	Learning Analytics cycle, as described by Clow <i>et al.</i> [15]	13
2.7	Triadic Certification Approach [14]	14
2.8	NPM trends comparison on microservices frameworks [3]	15
2.9	A cytoscape.js demo [18]	17
3.1	Creating and Editing Activity Diagram	23
3.2	Deployment Activity Diagram	24
3.3	Student experiences IAP Activity Diagram	25
3.4	Consult Analytics Activity Diagram	26
4.1	High Level Architecture Diagram	31
4.2	Components Diagram	33
4.3	Flow Diagram - Insert activity in IAP	37
4.4	Flow Diagram - Assign objective to IAP	38
4.5	Flow Diagram - Connect activities	39
4.6	Flow Diagram - IAP Deployment	40
4.7	Flow Diagram - Student Activity Deployment	41
5.1	Code organization on the backend repository	44
5.2	IAP Sample payload	46
5.3	Create activity request and response	47
5.4	Get activities for deployed IAP	48
5.5	Environment variables for the backend	48
5.6	Backend log output	49
5.7	Frontend Code Organization	50
5.8	Original PoC for the frontend requirements	51
5.9	Tablet view of the authoring tool	52
5.10	Frontend environment file	53
6.1	Unit test suite in Insomnia	57
6.2	Frontend Homepage	57
6.3	Deployed IAPs view	58
6.4	Deployed IAP Activities view	58
6.5	Registered Activities view	59

6.6	Activity Analytics view	59
6.7	Adding an activity to the IAP modal	60
6.8	Adding an activity to the IAP modal	61
6.9	IAP with some activities	61
6.10	Auto-formatted IAP	62
6.11	Activity Context Menu	62
6.12	Set objectives for IAP	63
6.13	IAP objectives and analytics view	64
6.14	The configuration page of one of Duarte's developed activities	65
6.15	Activity Deployment Diagram from Duarte	65
6.16	CHIC's authoring tool	66
6.17	CHIC's mobile application for tourists	68
A.1	Environment variables sample for the backend	76
A.2	Environment variables sample for the frontend	77
A.3	NGINX sample configuration	78

List of Tables

3.1	Functional Requirements	27
3.2	Non-Functional Requirements	27
6.1	Unit Tests	56

Abbreviations

FEUP	Faculty of Engineering of University of Porto
iLRN	Immersive Learning Research Network
Inven!RA	Inventive agency amidst Reticular ecosystems of Atopic habitats
IAP	Inventive Activities Plan
BEACONING	Breaking Educational bArriers with CONtextualized pervasive and gameful learnING
GLP	Gamified Lesson Plan
CHIC	Cooperative Holistic view on Internet and Content
LMS	Learning Management System
JSON	JavaScript Object Notation
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
URL	Uniform Resource Locator
API	Application Programming Interface
REST	REpresentational State Transfer
CRUD	Create, Read, Update and Delete
LBS	Location-Based Services
DOM	Document Object Model
UI	User Interface
UX	User eXperience
GUI	Graphical User Interface

Chapter 1

Introduction

Despite being a hot topic, immersive education continues to be a fascinating and stimulating challenge for whoever addresses it. In this work, the focus is to provide learning designers with the necessary tools to build experiences that they would be able to visualize and track, focusing on the foundations and backend application to facilitate it, as well as a proof of concept frontend. To do this, we will explain the novel concept of Inven!RA, analyse the needs we identified and how we propose to tackle them.

This document describes the main ideas that drove the development of the Inven!RA platform, how we generated and developed the prototype, and go through use cases that use the implemented prototype and the Inven!RA platform concepts.

This introductory chapter lays out the context, objectives, and challenges for this work, and provides a brief document structure overview.

1.1 Context

This work builds upon previous work done in the BEACONING¹ project, mainly its approach of having an authoring tool for learning designers producing Gamified Lesson Plans that can be deployed by teachers. These concepts have been rendered more flexible and open as detailed in Chapter 4. The BEACONING project is detailed in Chapter 2.

This dissertation was developed with advice from researchers of iLRN - the Immersive Learning Research Network², an international organization that is focused on developing the potential of immersive learning. iLRN's mission "is to invite and organize scientists, practitioners, organizations, and innovators from across the disciplines to explore, describe, showcase, and apply the optimal use of immersive environments for formal and informal educational purposes"[27]. One of this dissertation's supervisors, Prof. Leonel Morgado, from Universidade Aberta, is a board member of this network. Prof. Dennis Beck, from the University of Arkansas, who provided advice to this work as reported later, is also part of iLRN's board.

¹<https://beaconing.eu/>, last accessed on July 3, 2020

²<https://immersivelrn.org/>, last accessed on July 3, 2020

In addition, the novel educational goals that the project aims to support were the outcome of close cooperation with further third parties: researchers from project PRINT/“Digital Transformation and Humanities” at UNISINOS (Rio Grande do Sul, Brazil), which aims to manage inventive educational activities that blend traditional education with games and pervasive and ubiquitous learning; and researchers from the CHIC project at FEUP and INESC TEC, focused on the gamification of tourism experiences. These projects are ongoing and developed in tandem with this dissertation, contributing with requirements and leveraging the architecture developed and documented in this work, as well as its prototype backend implementation. A prototype frontend was also developed to test and demonstrate such features, and successful integration was accomplished with third party activities developed for our prototype.

1.2 Motivation and Objectives

The main motivation for this work was enabling a better environment for building, sharing, experiencing, and tracking inventive activities, as well as facilitating the interactions needed between the designers and leaders of such experiences and their target audiences.

The objectives that we laid consist on the specification and development of an architecture supporting authoring tools for Inventive Activity Plans (IAPs) and their deployment, focusing on the backend operations that will drive this and other future features of Inven!RA, but not disregarding the frontend functional requirements as proof of concept.

1.3 The Challenge

1.3.1 Overview

The management, deployment and tracking of educational or otherwise objective-oriented activities remains an unsolved problem, or feebly addressed. The challenge with Inven!RA was to provide a significant example of the future at-wild (or widespread) deployment of immersive education, by delivering a palpable and versatile solution that would provide real value to all involved.

Education is an obvious and intuitive example to follow, but our aim is not to get blinded by traditional classroom contexts, and provide a sufficiently broad architecture to tackle any non-conventional set of activities, with the common denominator being that there are objectives to achieve in a given set of activities and that tracking the progress of the users towards those objectives is a key aspect.

To achieve this, we devised a proof-of-concept architecture and prototype implementation that provides a concrete rendering of this core set of features.

A core challenge of this work was also the continuous integration with other teams and projects, occurring continuously while still defining the core concepts.

1.3.2 Research Questions

1.3.2.1 The impact of tooling and analytics on teaching

Learning designers – a term we will often use in this work to describe someone who designs a set of activities with the purpose of conveying knowledge or other objectives - often lack the necessary tools to visualize, deploy, and track their work.

Analytics, in particular, are of paramount relevance to better understand how one is progressing towards goals. In Alhadad *et al*[13], the authors state that “Learning analytics must account for key stakeholders in teaching and learning”, which is something Inven!RA takes to heart.

As noted by Costello *et al*.[16], institutions are often only using e-learning as a repository for uploading academic materials, instead of taking into consideration the user’s perceptions and needs. Moreover, even if those needs are met, the data is often skewed towards classic educational activities, with a wide range of more dynamic, collaborative and otherwise out of the box activities remaining hard to track.

In the progress of this work, and towards the subsequent progress of Inven!RA, the question on the impact of adequate tooling and data, as it pertains to online learning environments, is a constant quest that is tackled.

1.3.3 The importance of an open ecosystem

The rising awareness of data security and anonymity is a current major concern, both for platform providers and users. As can be extracted from Kaspersky’s³ Global Privacy Report of 2020, users are nowadays more prone to protect their information and apply additional security measures online[30]. While in the context of this work user management was not a priority, the solution was designed accounting for data anonymity, abstraction, and integration with legacy platforms in educational institutions: Learning Management Systems (LMS) such as Moodle.

The concept of the Inven!RA platform is based on mediating and improving the work of learning designers and other users by integrating activity deployment and tracking with other platforms and providers, acting as middleware where needed. By keeping the ecosystem open, as opposed to handling everything monolithically, existing and future infrastructures and activity can be leveraged to provide flexibility for activities. Individual activities for inclusion in plans are only registered on the system and configured for deployment, not necessarily provided by it.

Throughout, this work expands on the value of this approach, how it impacted the architecture’s design and development and what can be learned from it.

1.4 Structure

This document began by shortly introducing the challenges, objectives and context of the work, as well as the real-world problems that inspired it and the research questions that are tackled, in this

³Kaspersky is a leading company on the security and antivirus business - <https://www.kaspersky.com/>, last accessed on July 3, 2020

[Chapter 1](#); it will continue with a brief study on the state of the art in [Chapter 2](#); then the document describes the process taken to reach the solution specification, in [Chapter 3](#); details the proposed architectural solution, concepts, and functionality, in [Chapter 4](#); explores the developed prototype and details its development, in [Chapter 5](#); goes through validation and testing of the prototype, in [Chapter 6](#); and finally addresses conclusions, lessons learned, and future work, in [Chapter 7](#).

Chapter 2

Background

This chapter addresses the background of this work, diving into related work and the ideas that inspired it, such as the management of dynamic educational activities and the harvesting of learning analytics and their usage. It addresses in more detail the BEACONING project, upon which the Inven!RA platform builds, but also other related work, in order to provide some perspective and understand what's readily available on the market and the differentiating factors of Inven!RA. Lastly, it provides an overview of the technologies used for development of the prototype.

2.1 Managing dynamic educational activities

Education comes in a variety of ways. In the current era of shifting societal dynamics, supporting learning is all the more of a challenge, with varying and demanding critical-thinking goals, not just specific hard skills and facts.

Going beyond traditional educational activities, the idea of dynamic educational activities encompasses several learning ecosystems, from field work and laboratorial classes to modern concepts such as serious games. Managing a wide array of activities in a cohesive manner is a complex challenge for educational actors, and one to tackle with the usage of appropriate tooling, supporting learning design and analytics.

The diversity of contexts and platforms that can be integrated is such that an open ecosystem is paramount to the feasibility of this solution, something at the heart of this proposal.

Inventive learning, as a new philosophical stance on learning, brings a novel perspective on the challenge of learning for this shifting context, by positing that learners are not simply analysing problems or solving problems. Deep, meaningful learning is actively about creating problems. For instance, when one tackles a reading, one generates a problem for the learning arising from that reading: a vision, a purpose, an intent, a context. Learning is about inventing problems and, in that process, inventing meaning, or as the original proponent of the concept stated, inventing the world[31].

The concept of an inventive activity thus places a great challenge for educational support tools, since more than just predetermined courses of action, diverse aspects need to be tracked, depending

on the actual goals that the educator aims to pursue. This flexibility is at the very centre of the Inven!RA platform, being reflected in its very name – Inven!RA stands for Inventive Reticular Atopic ecosystems. As proposed by Schlemmer et al.[20], Inven!RA is a “proposal framing novel platforms that view teaching and learning as a path intertwining different ecologies of participants, both human and non-human, within overlaid habitats of atopic and network epistemologies, where innovation and education emerge as ecosystem comprehension”.

2.2 Related Work

The following section analyses related work, with a focus on BEACONING and its authoring tool.

2.2.1 BEACONING

BEACONING[21] was an European project. Its acronym stands for “Breaking Educational Barriers with Contextualised, Pervasive and Gameful Learning”. It started in January of 2016 and ran for 36 months, with 15 institutional partners in 9 different countries across Europe. This award-winning project[2] had 3 main goals:

- Integrate technological, pedagogical and social perspectives using pervasive, context-aware and gamified approaches ensuring that the BEACONING platform engages a community of learners;
- Develop, implement and validate a platform that utilizes gamification, pervasive gaming, procedural game content generation, game authoring, human-computer interfaces, learning analytics and a problem-based learning model and that is usable, adaptable, extendable and sustainable;
- Explore and measure the level of engagement, effectiveness and impact that the BEACONING platform provides towards incentivising learners and promote acquisition and transfer of knowledge and skills, and also validate and provide an exploitation and business plan for the platform adoption.

In the context of Inven!RA, it is useful to look at BEACONING’s authoring tool, architecture, and the concept of GLPs – Gamified Lesson Plans. For this, the analysis focused on the projects deliverables, readily available online at BEACONING’s deliverables page¹, namely D3.6 – System Architecture[28], D4.4 – Prototype[24], and D4.8 – GLPs[34]. The analysis also checked the BEACONING authoring tool code and its main developer, alongside the dissertation work it spawned[22].

The next sub-sections go through what was learned from these documents and influenced the Inven!RA platform.

¹<https://beaconing.eu/downloads/>, last accessed on July 3, 2020

2.2.1.1 Gamified Lesson Plans - GLPs

The notion of Gamified Lesson Plans, or GLPs, is at the very core of the BEACONING project and experience.

BEACONING aimed to break barriers that "constrain learning from a social, spatial and technological point of view, enabling teachers to co-design GLPs immersed in the real world and supported by a broad community"[34], as well as facilitating students ways to learn through these lesson plans in an immersive, interactive, game-like form, and by facilitating these experiences in a variety of devices and contexts.

The concept of a GLP is about closely intertwining learning objectives and a game plot, generating a plan to achieve those objectives. These GLPs integrate a number of activities in specific parts of the game plot that enable it, which can be mundane tasks or entire mini games. In BEACONING there is a strong focus on gamification, and therefore these activities are usually mini games.

At its core, the GLP is the plan the student will undertake to achieve the learning objectives. It is represented programmatically by a graph, comprising activities that can be instantiated, and the configuration of analytics to be collected to track progress towards the learning objectives.

Figure 2.1 shows the GLP data model, which provides insight into how these concepts connect with the BEACONING backend. Studying this model and the decisions behind it allowed us to extract some of the data we would also need in our respective data model, but most importantly we focused on the entities and how we could simplify them, for example by fusing nodes with activities and handling optional fields as an array of properties, to make the mandatory elements stand out.

2.2.1.2 BEACONING's Authoring Tool

BEACONING's Authoring Tool is the component that enables learning designers to author, edit and deploy Gamified Lesson Plans according to their specific needs. This software enables them to create Missions and Quests, which are the structure of the learning objectives, where a set of missions constitutes a quest.

Figure 2.2 shows the tool's GUI in its final form. This user interface was designed with the purpose of being intuitive for a Learning Designer (which can be an educational content developer or an experienced teacher) focusing on a hierarchical view of the Gamified Lesson Plan. The integration with the game plots is achieved by a graph-based description that depicts the unfolding of the underlying narrative.

From Figure 2.2, one can extract some of the core principles and features available on BEACONING's authoring tool. On the left, there is a button to add quests or missions, as well as saving, cloning, and creating a new GLP. On the right, there are minigames, which can be dragged and dropped into the plan, on the centre. On the plan, one can see connections between the activities, as well as some data related to each one.

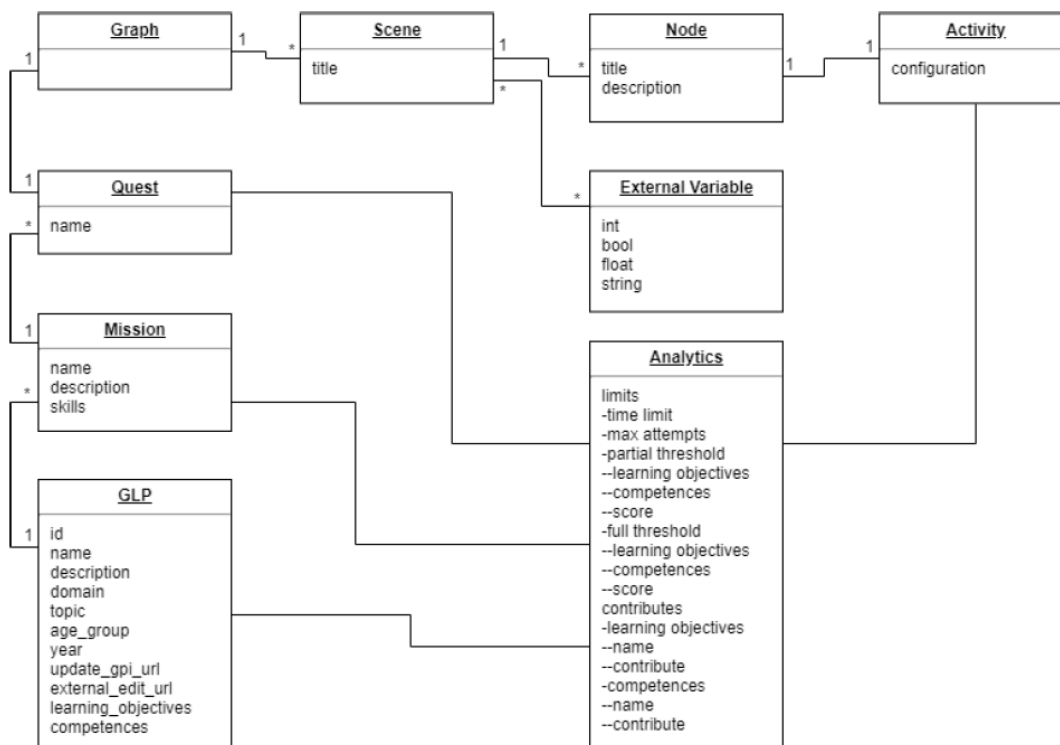


Figure 2.1: GLP Data Model proposed in [34]

The tool allows the user to edit several of the metadata available, usually through the pop-up of a modal-like editing form. Figure 2.3 shows what the edit screen for an activity looks like.

After a GLP is finalized and a teacher has added the quests, the GLP is ready to be deployed by a teacher to a target audience, for instance a class of students. This is done by integrating the backend with Learning Management Systems of the teacher and its class.

2.2.2 Others

This section explores some other applications related to the objectives, with the goal of understanding what the market provides for someone on the lookout for a technological approach to narrative plots for learning.

2.2.2.1 Twine

Twine² is a free and open-source tool for "telling interactive, nonlinear stories in the form of web pages"[7]. It is available for all major operating systems, therefore running locally. A version 2.0 is available as a web application written in JavaScript and is the current version being maintained by a community on GitHub[32], with over 10 active main contributors.

²<https://twinery.org/>, last accessed on July 3, 2020

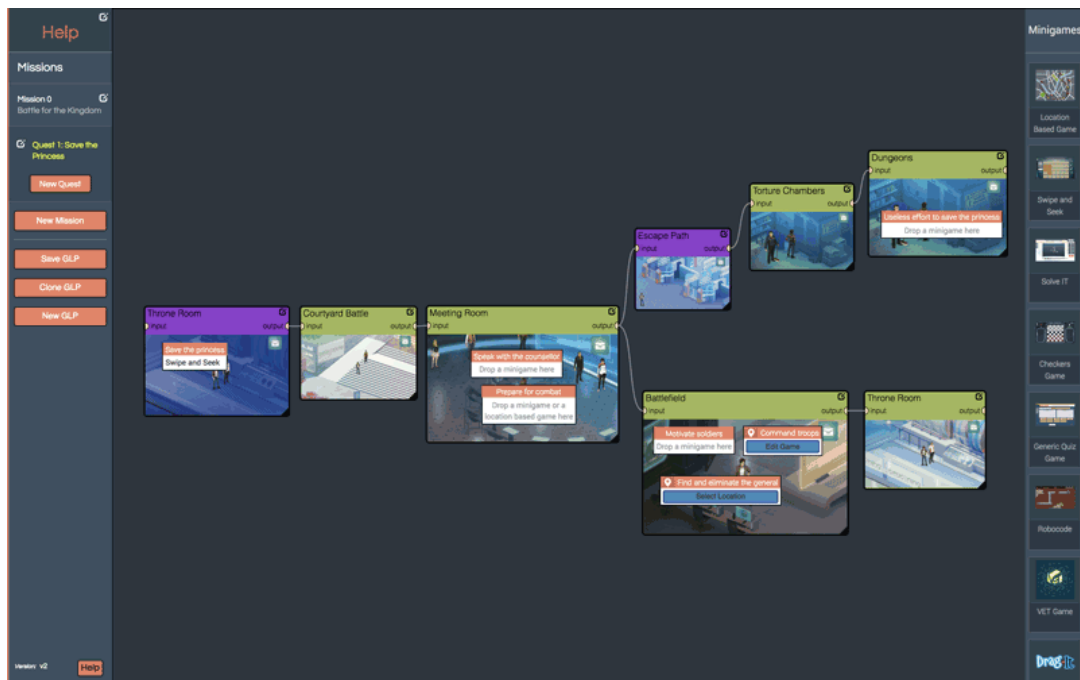


Figure 2.2: BEACONING's Authoring Tool, extracted from [1]

Twine is, at its core, a tool for creating hypertext. The branching possibilities allow the user to have some measure of agency over the experience. Much of Twine is dedicated to helping the designer at keeping track of the stories' structure visually with a story map.

What drove us to study Twine was its simple but flexible editing tool, from which we can gather inspiration. However, when researching, we found a vibrant community of designers and a tool that can be used for abstraction on game making, and some interesting concepts such as how story formats are handled[8].

2.2.2.2 Padlet

Padlet³ shares some of Twine's DNA, however being more oriented to education. It is a popular tool amongst researchers, producing several hits when searching digital libraries such as ACM.

Padlet is software used to make and share content, somewhere between a doc and a full-fledged website builder, "Padlet empowers everyone to make the content they want"[12]. It has an intuitive user interface for creating content, it is collaborative, flexible, and portable.

The end result of a Padlet instance consists of a virtual wall, where information, audio, images, video and a myriad of other files can be shared and connected. The creator of a wall has control over the content, design, layout, and privacy of it, such as who has access to the wall.

In a study by Deni *et al.*[19], it was concluded that in activities using Padlet, students valued the flexibility given to them in completing the tasks on Padlet: "We can answer if we have time or browse and learn from others' answers if we don't have time to answer."

³<https://padlet.com/>, last accessed on July 3, 2020

Figure 2.3: Adding an activity on BEACONING, extracted from [1]

Our main takeaways from Padlet are its editing tool and the impressive integration agnostic of file types, to provide a narrative that is easy to follow as a learner. The deployment and management of experiences through the likes of URLs, QR codes, and privacy management on the part of the designer, were also highlights of our experience with the software.

2.3 Analytics harvesting

Analytics play a big part on the way Inven!RA, its authoring tool and how its activity plans unravel. In the context of our work, an analytic is anything that can be tracked, weighed, and carry some value towards an objective, either discrete or qualitative. Learning analytics is "a fast-growing area of Technology-Enhanced Learning (TEL) research, with strong roots in a variety of fields and approach by a range of perspectives"[23]. They can be defined as the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs.

If you take the example of a traditional school exam done online, you can extract that each question provides a value depending on the input provided by the student and contribute towards the goal of achieving a grade. Moving to a video game, a specific action can contribute towards an in-game objective, and when the player performs that action, that can trigger a trackable analytic.

Gathering this data might be complex, and it is something we will address in our prototype and specification. By involving the designer and providing integration to developers, we can track analytics and associate them with a plan of activities. Upon deployment, we can then provide this useful data to the designer.

By having access to this data in a closed feedback cycle, the possibilities are numerous and can allow designers to improve the way they design their experiences, and most importantly help

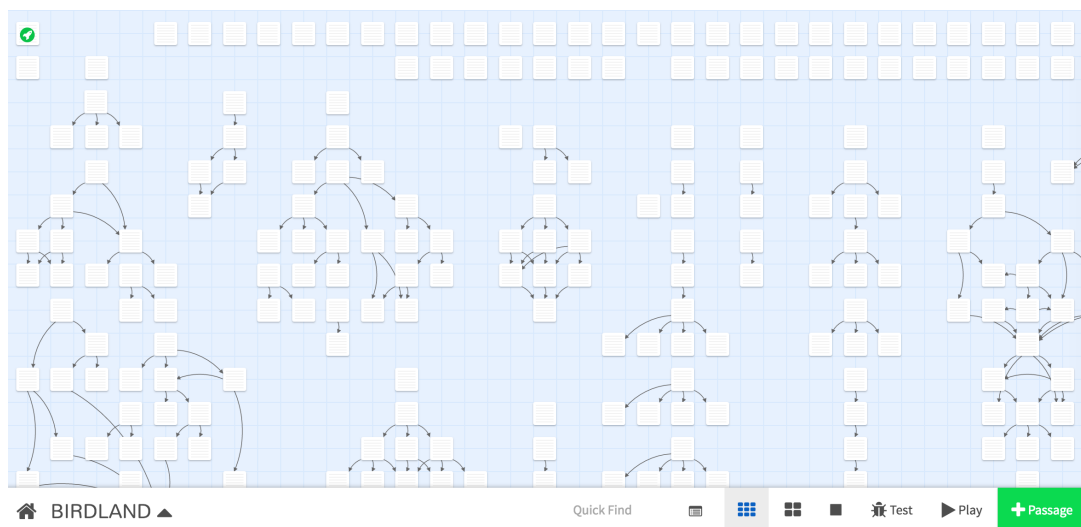


Figure 2.4: Twine 2 Story Map [7]

them identify the learners falling behind and what is causing it, giving the instructor effective data with which to act upon in improving the learning experience.

Harvesting this data can be tackled in different ways, with varying levels of complexity. For Inven!RA, at this point, we rely on the activity providers for this data, which we only propose to relate to the users, in complete obfuscation to the provider, acting as middleware between the activity provider and the LMS, and relay them to the educator.

Having data is useful, but we feel like what really sets our solution apart is the association and abstraction of these analytics, by associating them to learning objectives. This way, we can quantify the completion of a learning task and investigate the detail, but also abstract from it if we want, when we only want to observe if the competence was attained. To better get the background of how researchers look at the certification of competences, we will present some study on this topic in the next section.

2.3.1 Approaching the certification of competences

To better design and implement plans of activities and the concepts of analytics and learning objectives, we will take a brief look at how researchers approach the problem of certification of competences.

The certification of competences is an important pillar of the Inven!RA concept, albeit it not being a major focus on this specific prototype for Inven!RA's authoring tool, we still tried to address it in the prototype, and provide ideas for its expansion in the future, laying out some seeds for future work.

With a focus on learning environments, and with the concept of Serious Games (SG) in mind, approaching completion and certification of certain competences is a tricky issue. As

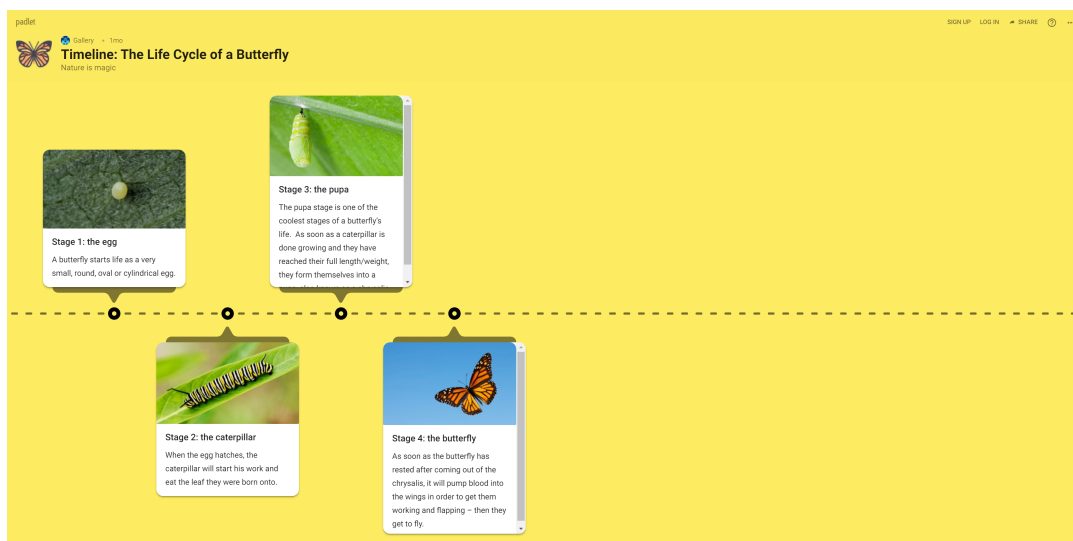


Figure 2.5: Padlet Sample Narrative [12]

stated by Serrano-Laguna *et al.* in their study on the application of learning analytics in educational videogames, this application “is hard to generalize, expensive to scale and difficult to organize”[37]. They propose a two-layered approach, with a first layer of generalization, and a second layer that generates reports tapping into those generic traces to gain insight into what the learner is playing and learning.

We find that the concept of training levels associated with challenges, as described by Baptista *et al.*[14] in the Triadic Certification Approach, visible in Figure 2.7, is a promising approach. This model combines competences with mechanics and enables the idea of different training levels. In this model, it is possible to adapt the challenges to different learning levels, which could provide an interesting solution to a diversity of students, as levels could be raised or lowered with student feedback in consideration.

2.4 Technologies overview

Before delving into architectural decisions, it became clear from our objectives that we will require a backend to provide a REST API, and a frontend web application for authoring and validating the backend functionalities. Easy setup, integration, extensive documentation, and easy expansibility are major factors in our choices, as well as the author’s own experience and industry usage, to ensure easy future maintainability and expansion of the solution.

In this section, we will provide an overview of the major selected technologies and why we thought they were the best choices for our prototype.

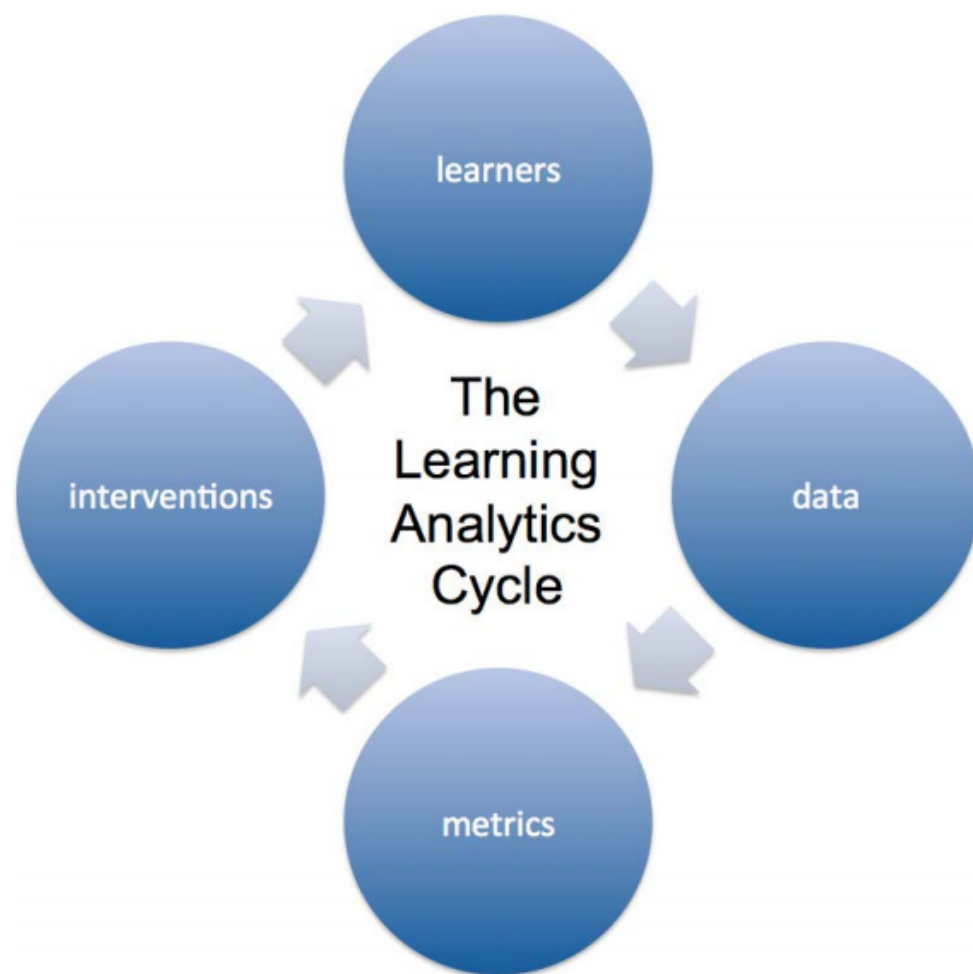


Figure 2.6: Learning Analytics cycle, as described by Clow *et al.*[15]

2.4.1 Backend

For the backend, we wanted to focus on microservices and on industry standards. We chose to base our implementation in Node.js, with a MongoDB database for data storage. We then expanded with Moleculer, a progressive microservices framework for Node.js, for intuitive microservices management and abstraction.

2.4.1.1 MongoDB

MongoDB⁴ is a NoSQL database, which is "a non-relational type of database with a flexible schema"[11]. It is document-based, which means it stores data in JSON-like documents[33].

Its intuitive way to store data, its easy integration with JavaScript and JSON based applications, as well as the flexibility of its non-relational nature were the main attractions to us.

By using MongoDB, we didn't need to worry too much early on about data structure, which enabled us to get a prototype up and running while we still were figuring things out, as well as

⁴<https://www.mongodb.com/>, last accessed on July 3, 2020

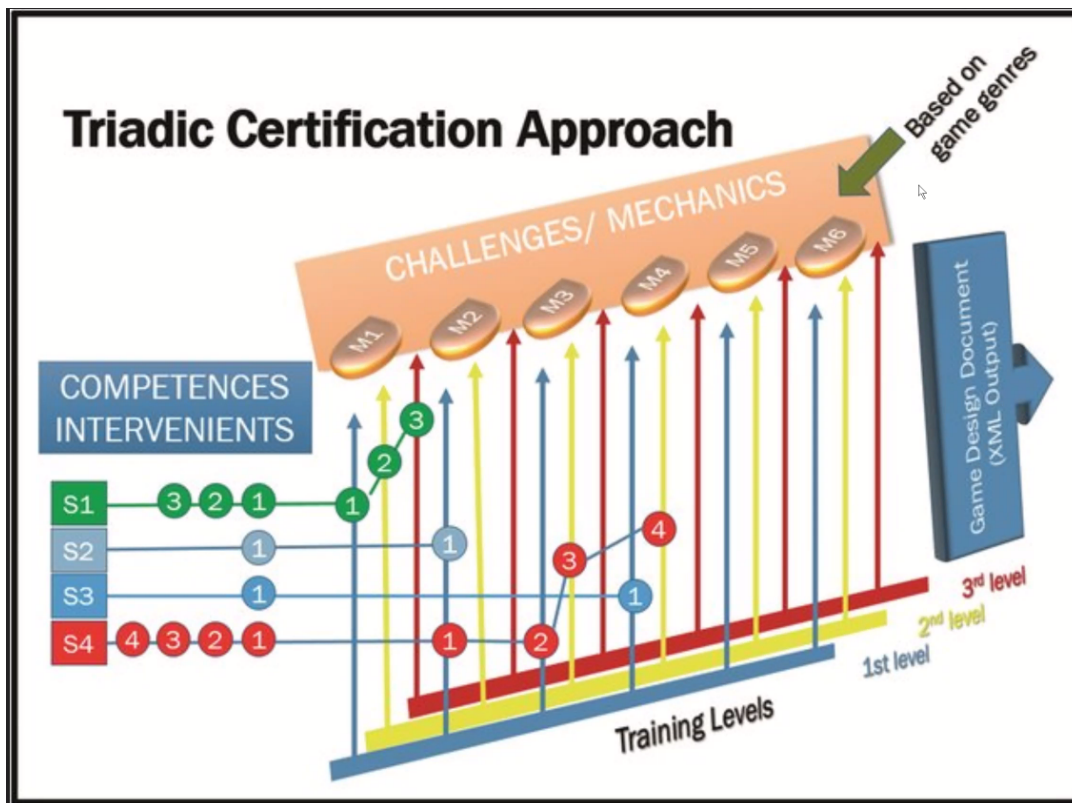


Figure 2.7: Triadic Certification Approach [14]

maintaining integrations with the other teams without breaking functionality when schemas didn't fully align. Its schema-less approach also enables us to adjust internal schemas as needed and it negates any need for schema migration at a database level.

In sum, MongoDB was an ideal choice for fast prototyping and flexibility on data storage.

2.4.1.2 Node.js

Node.js⁵ is one of the most popular runtimes for backend web applications, boasting a 49.6% adoption by developers on the 2018 Stack Overflow Developer Survey[6]. It "is an opensource, asynchronous event-driven JavaScript runtime designed to build scalable network applications"[35].

Given the author's previous experience with the technology, its industry-wide acceptance, wide community support and documentation, as well as its performance and versatility through modules, the choice was easy. By developing the backend in a JavaScript based language, we also have the opportunity to produce both the backend and the frontend using a single language.

Another main factor in this choice was Node.js's cross-platform nature and easy deployment, as well as industry leading performance and usage.

⁵<https://nodejs.org/en/>, last accessed on July 3, 2020



Figure 2.8: NPM trends comparison on microservices frameworks [3]

2.4.1.3 Molecular

Getting in more concrete territory, we wanted to find a framework to organize and manage microservices. After prototyping with Seneca, Devis and Molecular, we went with Molecular⁶. Its industry acceptance and developer support also helped in this decision, as can be seen in NPM trends[3], illustrated on Figure 2.8.

Molecular is an open source progressive microservices framework for Node.js, with several built-in modules for valuable features we wanted to use, such as caching, fault handling, middleware, asynchronous promise-based functionality, pluggable loggers, transporters and serializers, and more[10].

By using Molecular and adhering to its format, it helped us identify and improve the way in which the code was organized and structured.

2.4.2 Frontend

For our prototype authoring tool, we wanted a web application that was easy to set-up and provided us the necessary tools for interaction with the backend. We chose to pick the React framework and use Bootstrap to quickly build up a user interface.

The other major choice we had to make was what to use to enable user interactions with the canvas, given that we wanted drag and drop support, and graph visualization and editing. We picked Cytoscape, as we felt it was the most extensive, versatile, and widely used platform for the effect. Below, we will provide an overview of each choice and why we picked them.

⁶<https://molecular.services/>, last accessed on July 3, 2020

2.4.2.1 React.js

React⁷ is a popular JavaScript library for building user interfaces[38], with a focus on single-page applications. It is an open-source library maintained by Facebook and a community of individual developers and companies. This library is popular for being declarative and component based.

The main purpose of React is to be fast, scalable and simple. Instead of using regular JavaScript for templating, it makes use of JSX, which is JavaScript that allows HTML quoting and uses it to render subcomponents.

React has built-in stateful components, which interests us as to avoid unnecessary API calls and keep information in memory.

Another main feature of React that piqued our interest was the ability to easily turn an application into a Progressive Web App, a concept growing in adoption in contrast to mobile apps[25]. Progressive Web Apps are built and enhanced with modern APIs to deliver native-like capabilities, reliability and installability while reaching anyone, anywhere, on any device with a single code-base and no installation[9]. This is relevant to us as it means we can have an application that runs well in mobile devices without additional development.

2.4.2.2 Bootstrap

To easily set up an appealing user interface and since the frontend was not the focus of our work and dedicated components were not a priority, we used the popular Bootstrap Toolkit[29]. Bootstrap is an open-source framework for designing web applications using HTML, CSS and JavaScript, which provides pre-built modules to use and customize.

Since we are using React as our main framework on the frontend, we picket the React Bootstrap framework, which adapts Bootstrap's JavaScript to use true React components without dependencies such as jQuery[4]. This choice allowed us to import individual components, as opposed to the entire Bootstrap library, and also to keep the code more neatly organized and using React's conventions on components, while using the full set of features that Bootstrap enables, such as a grid system and several useful UI components.

2.4.2.3 Cytoscape

Cytoscape⁸ is an open source software platform for "visualizing complex networks and integrating these with any type of attribute data"[36]. It was originally designed for biological research but is now a general platform for complex network analysis and visualization. Cytoscape's "core distribution provides a basic set of features for data integration, analysis, and visualization"[18], and additional features are available as plugins.

While there are several versions of Cytoscape, we use the JavaScript library, which allows to easily display and manipulative rich, interactive graphs. Since it is written in JavaScript and quite

⁷<https://reactjs.org/>, last accessed on July 3, 2020

⁸<https://cytoscape.org/>, last accessed on July 3, 2020

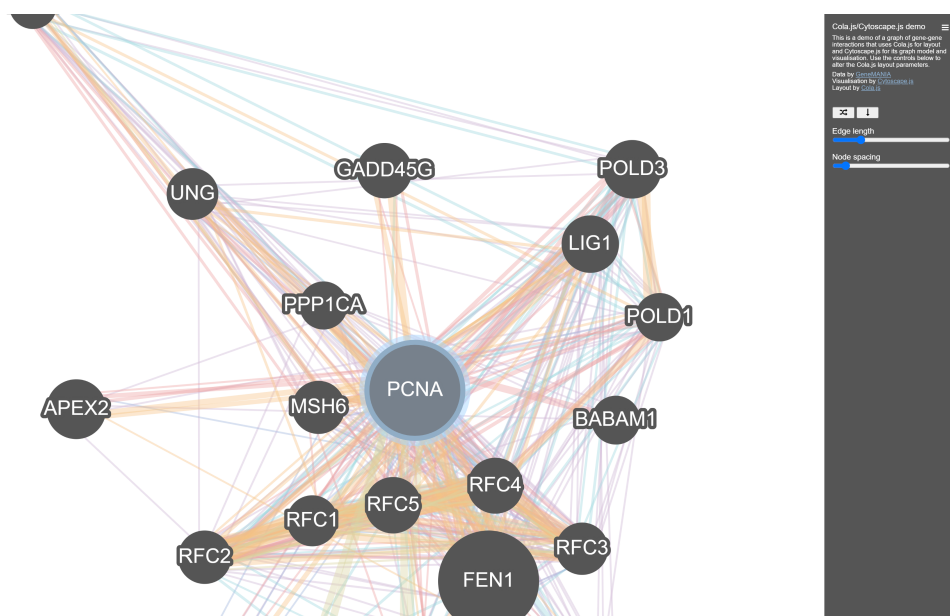


Figure 2.9: A cytoscape.js demo [18]

mature, it allows the user to interact with the graph and provides hooks for several user events, which we can use[26].

Once again attending to our choice of React, we found the react-cytoscape package[5], which allows us to use Cytoscape as a React component, for easier and clearer integration.

2.5 Summary

In this chapter we laid an overview of the background and relevant related work, both on the concepts and on implementation. We took a focused approach on the work that most inspired us and better represents what is envisioned by the Inven!RA platform and its objectives, in the form of the BEACONING project. Lastly, we gave an overview of the technologies used in the prototype developed.

Chapter 3

Getting to the solution

After addressing the necessary introductions and state of the art, we will document the process by which we achieved the proposed solution for this work. For this undertaking, we leveraged contacts with industry experts with interest or otherwise involved in the Inven!RA project, and from different but related backgrounds. With said contacts, we worked towards user stories and activity diagrams from which to extract functionality, which we will document in this chapter, by describing functionalities from a user perspective.

3.1 How we raised requirements

Before getting to work, we tried to better understand the needs and wants of the people who would be interested in the platform. For this, we made use of the partnerships previously established and briefly mentioned in Chapter 1, and collaborated with foreign researchers involved with the Inven!RA platform and the iLRN, with experience and interest in pervasive gaming, gamification, learning and education; notably Dennis Beck, from the University of Arkansas, USA, and Eliane Schlemmer, from UNISINOS University in Rio Grande, Brazil.

Along with this work's supervisor Leonel Morgado, Dennis Beck is a board member of the iLRN, a Professor at the University of Arkansas, and also a consultant for an online high school in the United States of America, with an interest in incorporating immersive environments by using gamified lesson plans. Eliane Schlemmer is a Brazilian Professor and researcher in the project PRINT "Transformação Digital"[39] at UNISINOS, which aims to develop innovative pedagogic activities, as well as blend traditional education with games and pervasive & ubiquitous learning.

In the sessions we shared, we discussed and collaboratively designed user stories for the project, in addition to getting to know previous and current projects they were working on. With the user stories and some useful concepts extracted, we then proceeded to design some activity diagrams from which to extract desirable functionality and to more clearly delineate the features we would need for a viable platform and what we would require to achieve it.

3.2 User Stories

Starting by the user stories, we let the interviewees speak for themselves in terms that would make sense to them, and just oriented where needed. By using an informal, natural language description, we let the users abstract from system concepts and focus on their vision. After a few sessions of listening and trimming the information, we ended up with 6 major user stories and 3 user types, as follows:

- Learning Designer – Creates an experience
- Teacher – Deploys an experience to an audience
- Student – Experiences and interacts with the experience

For a better visualization, we created the following personas based on the user types:

- Sue - Sue is a Learning Designer. She is interested in making sure information is reaching students and teachers clearly, enabling teacher's awareness and decision-making ability, providing tracking of how students are participating in the activities and how that participation contributes to the overall learning objectives
- Alice – Alice is also a Learning Designer, but one with a passion for escape rooms and narratives
- Mike – Mike is a Teacher with a passion for technology and improving teaching with modern solutions
- Anna – Anna is a student in an online school

We will now do a brief description of the user stories we designed, by going through the actions relevant for each one. In these user stories, we will start focusing on the notion of IAP – Inventive Activity Plan, which is a similar concept to BEACONING's Gamified Lesson Plans, previously addressed on [2.2.1.1](#), but adapted to Inven!RA and abstracted from the concept of gamification, albeit still compatible with it.

On the next subsections, we will present the 6 major user stories worked on with the interviewees.

3.2.1 US01 - Learning designer creates a linear IAP

Sue is interested in students filling in a form about field activities before doing them, and making sure students learned about how to do that. So, she goes to the Inven!RA Authoring Tool and chooses a linear IAP template.

She drags a pre-test activity into it. Then a reading activity. Then a form-filling activity and a sync/async discussion activity, and finally, a field report upload activity. She then connects all of these.

She enters descriptive labels for these activities in the authoring tool.

She opens the reading activity, and inside it checks the options available for data collection and analysis. She selects “download performed for each reading”.

She opens the form-filling activity and checks the setup options. She enters a URL for an online form (e.g.: Google Forms) and a URL to the resulting spreadsheet. She knows that this way the completion process will be available to the Inven!RA platform.

She opens the sync/async discussion activity and sets it up similarly. She then moves onto the field report and does similarly – the field report is a combination of form URL and cloud file folder URL or some online portfolio tool.

Then she needs to link this flow to learning outcomes. So, she “flips” the IAP and she sees all the activities in rows, and several empty columns.

She enters skill labels in columns, and clicks a cell: row “form-filling”, column “field procedures”. A skill card opens. She has a field to enter a reporting function and a list of all available data for that activity (extracted from the actual setup URL). She selects some of the fields to provide a % outcome towards “field procedures”. She does similarly for other cells.

Saving, her IAP is now ready to be deployed by teachers.

3.2.2 US02 - Learning designer creates a non-linear IAP: Escape Room

Alice wants to create an IAP for her escape room business.

She begins by introducing the activity “Narrative information”. Then, she names it “Read the initial narrative”.

She imagines how the activity would play out: the students begin by hearing or reading the narrative and they would search for clues in text files shared in a Google Drive (or similar service) folder. So, she drags to her IAP the activity “Read documentation” and calls it “Find clues”. She then adds an “inventory” activity, which she names “catalogue clues”.

Thinking about the fun dynamic that would ensue, by looking for clues and sharing in the previously set up WhatsApp (Messaging Service) group, she reflects this in the IAP by dragging an “Answer” activity, that she then names “Find the clue”. At this point, she realizes there are multiple clues and there will be differing paths, according to what clue is found. The clue can be a rabbit, a hatter, and so on. She imagines a chat will happen with each clue on WhatsApp. So, she drags several “Chat” activities and names them according to each clue.

For clue number one, the rabbit, Alice knows that the next step will be posting on the WhatsApp group some activity in the city. So, she adds a “Publish on WhatsApp” activity.

At this point, the student will be on a loop until he finds the clue, with successive help from the teacher.

When Alice is satisfied with the flow, she saves the IAP, which will be ready for deployment to a given audience.

3.2.3 US03 - Learning designer edits previously saved IAP

Alice is a busy business owner and could not finish her IAP in a single sitting. She saved her IAP when she last worked on it.

To get back into action, she simply accesses the Inven!RA Authoring Tool and navigates to “Saved IAPs”. There, she can find a list of all previously saved IAPs.

Alice wants to continue her Escape Room IAP, so she clicks “Escape Room IAP” from the list and is forwarded to the editing screen. There, she can get right back into designing her experience.

3.2.4 US04 – Teacher deploys IAP to a class

Mike’s school started using Inven!RA and has some previously configured IAPs ready to deploy. Mike wants to try it with one of his classes.

To do this, he accesses the Inven!RA authoring tool and clicks “Deploy IAP”. He then selects an IAP from the list of available ones. A prompt to select his target audience is shown. He selects the wanted class, which matches the classes he had set up on the school’s Moodle platform, and clicks “Deploy”.

Mike is given some instructions on how to share the activities with his class and how to track the activities in the platform.

3.2.5 US05 – Student experiences IAP

Anna is a student on an online school. She received an email mentioning there was a new activity to complete for her programming class, and she will be using Inven!RA.

She accesses the school platform or the Inven!RA platform and goes to her IAPs. There, IAPs which she has access to are available. She selects the IAP mentioned in the email and upon clicking “Start” is forwarded to the first activity.

3.2.6 US06 – Teacher tracks the audience’s progress on a given IAP

After deploying an IAP to his class a few days ago, Mike wants to see how they are doing. To do this, he accesses the platform and goes to “Running IAPs”.

In the screen, Mike can select from a list of currently deployed IAPs. On each, a shortcut for analytics is available. He clicks the one he wants to track and is forwarded to a screen which shows him all the objectives and where each student stands.

3.3 Activity Diagrams

In this section we will present some activity diagrams, as well as a simple description, formalizing the information extracted from the user stories, for the major applicational flows identified. Since these diagrams were designed with the same team as the user stories and with the purpose to raise

requirements, the level of abstraction is high, only taking into consideration Inven!RA as a single system capable of providing user actions.

3.3.1 Creating and Editing Activity

This is the main applicational loop for the user, wherein the learning designer is working on an IAP, by configuring the necessary artefacts in the system. This loop can be extracted from user stories US01, US02 and US03.

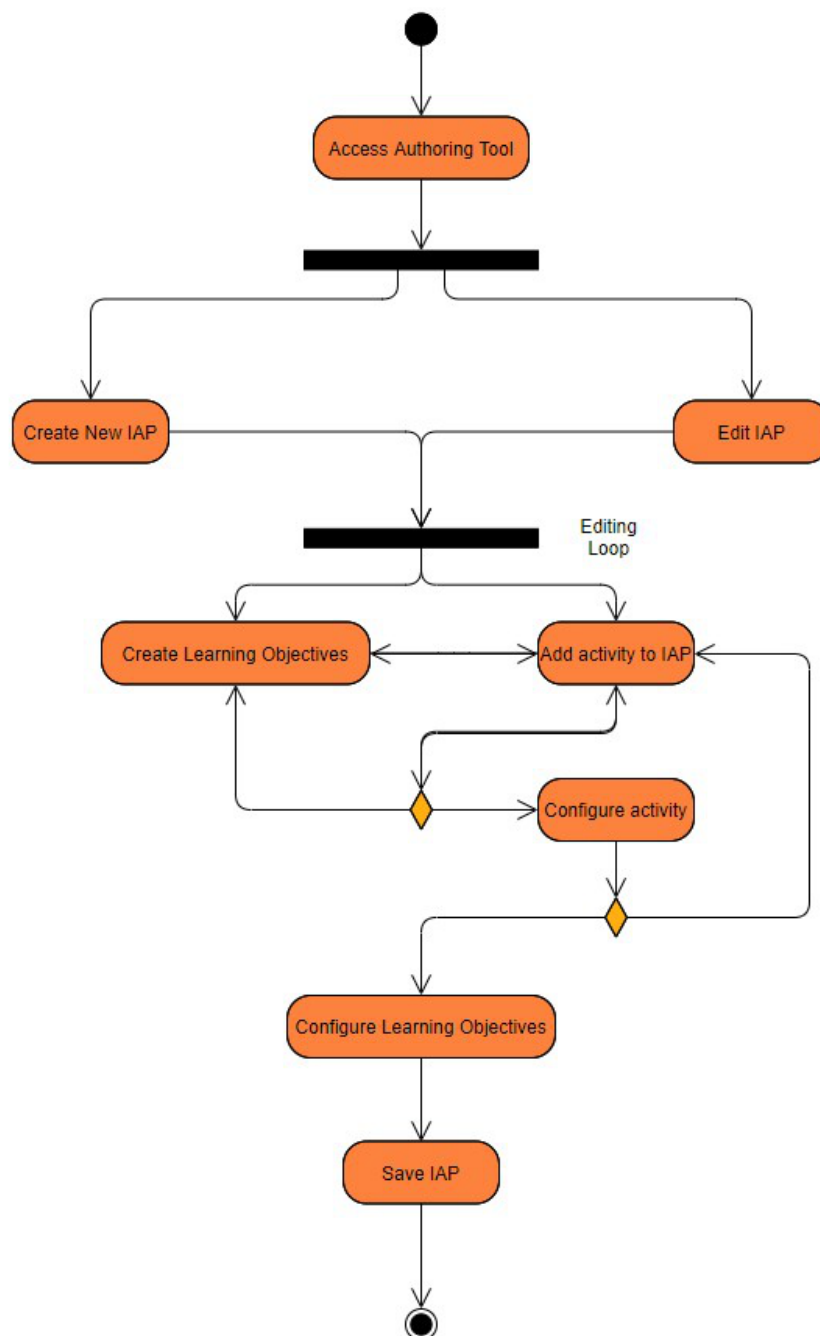


Figure 3.1: Creating and Editing Activity Diagram

3.3.2 Deploying Activity

In this activity, extracted for user story US04, the teacher selects an IAP, previously created by a learning designer, and deploys it to a target audience, which can, for example, be a classroom or any other subset of learners.

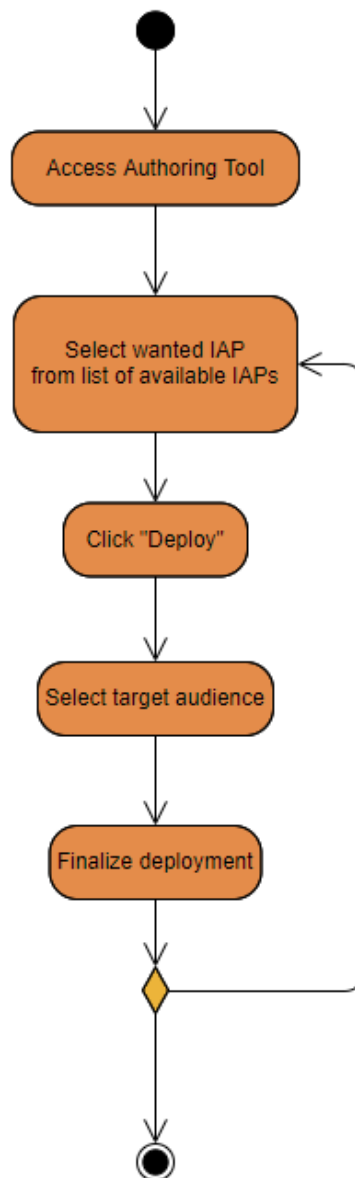


Figure 3.2: Deployment Activity Diagram

3.3.3 Student experiences IAP Activity

In this simple activity, regarding user story US05, the action starts with a notification to the user, which can be anything from an email to an in-class interaction, or even a notification on the LMS in use. Note that in the diagram we do not mention a learner UI to check available IAPs, as this was considered out of scope for this work and was simplified.

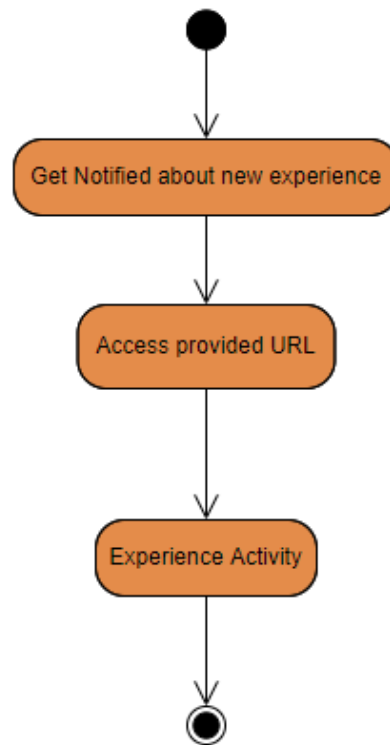


Figure 3.3: Student experiences IAP Activity Diagram

3.3.4 Tracking IAP Analytics Activity

For this activity, the teacher wants to track the progress of the previously deployed IAP, at any given time, by consulting the available analytics on the system.

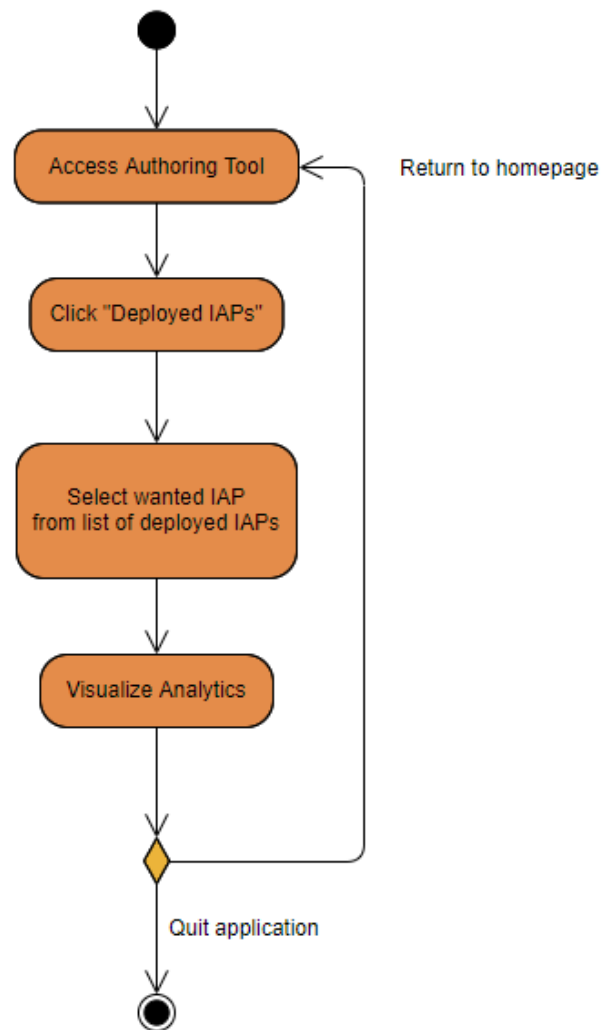


Figure 3.4: Consult Analytics Activity Diagram

3.4 System Requirements

After we got to a place where we felt satisfied with the ideas, stories and flows designed with the team, we reviewed the challenges and research so far, in order to formalize the requirements for the overall system, identifying the most crucial functional and non-functional requirements, summarized on Table 3.1 and 3.2.

Requirement	Description
IAP manipulation	The system must provide an interface wherein a user can create or edit an Inventive Activity Plan
Insert activities into system	The system must be capable of supporting a myriad of activities, which can be inserted in the system by an open API provided to activity providers
Configure activities in IAP	While editing a IAP, the user must be capable of configuring each activity, in accordance with the information provided during activity configuration
Configure learning objectives in IAP	Designer must be able to associate analytics with learning objectives of an IAP
Teachers must be able to deploy an IAP	As a teacher, I must be able to select from the list of available IAPs and deploy the one I want to a set of target users
Teacher must be able to consult analytics	After deploying an IAP, the teacher must be able to consult the analytics for it

Table 3.1: Functional Requirements

Requirement	Description
System modularity	The system must be designed in a way that promotes extending functionality by adding new pluggable modules
Cross-Platform	The system must maximize compatibility with a wide array of devices by using industry standards

Table 3.2: Non-Functional Requirements

Chapter 4

Proposed Solution

The following chapter documents and clarifies the relevant concepts for the solution, the architecture, design choices and otherwise relevant decisions in the development of the platform. We will begin by describing the concept and terminology moving forward, followed by an high level description and then getting into the architecture. Finally, we will present some applicational flows in the form of sequence diagrams.

In this chapter we will address the authoring tool application as a whole, but it is important to make the distinction that the backend is a standalone program, designed to be generic and expandible within the Inven!RA platform and not only this authoring tool. It can be, and is, used by other implementations and purposes, but for the scope of this work we focus on the authoring tool.

4.1 Concept

The Inven!RA platform proposes to tackle the challenges described on Section 1.3, and more, by providing a medium for Inventive agency amidst Reticular ecosystems of Atopic habitats, within which knowledge emerges[20]. In this work, however, we will focus on the challenges already described, as well as on the requirements raised on Chapter 3, which by themselves focus on the authoring tool part of the Inven!RA platform, which is the star of this study.

During the conceptualization phase, it became clear that the main goal of this work was the conceptualization and initial development of the backend services to enable Inven!RA's vision. This is essential to the continuation of the work, as architecture and payloads are addressed, to enable a solid foundation for future work. In order to test all of this and to gather useful feedback, however, a frontend is required. So, we approached the project backend-first and used the frontend to explain and prove its viability, as well as validate it with users.

Inven!RA's authoring tool, then, follows in the footsteps of BEACONING's counterpart. We leveraged all the lessons learned there, and proven concepts such as GLPs, to design the architecture.

4.2 Useful terminology

Before getting into the architecture, we think it is important to lay out some concepts, as they pertain to the Inven!RA project and its Authoring Tool.

IAP - An **Inventive Activity Plan** is a graph of connected or independent activities that contribute towards learning objectives.

The IAP can be akin to a lesson plan or a narrative to follow in order to achieve one or several goals.

Activity – In this context, an activity is a generic notion for something the user can do and which needs to be tracked by the teacher.

As it pertains to an IAP, an activity is a node in the graph. Each activity has its own configuration details and set of available analytics.

Learning Objectives - Each IAP has a set of learning objectives, towards which the activities contribute.

The relationship between each activity and the learning objectives is a formulaic relationship or algorithm mapping that activity's available analytics to the objectives.

Each learning objective is structured into tiers or levels.

To better illustrate the use cases and their interactions, we will consider the following user types:

- Third-party activity provider
 - Designs an activity, registers it with the backend, and hosts the activity and its analytics
- Learning designer
 - Creates an IAP
 - Associates analytics with the learning objectives
- Teacher
 - Deploys an IAP
 - Assigns an IAP to users (such as a class)
 - Monitors the progress of deployed IAPs towards their learning objectives
- Student
 - Experiences the IAP and interacts with it, generating analytics with its interactions

4.3 High Level Description

On a macro level, we followed a familiar and simple design, leveraging the standards of today's web applications. The brains of the operation is the backend, a Node.js standalone application that exposes a REST interface. This backend is linked to a NoSQL database, in this case we chose MongoDB, which is where we store all necessary information.

The system is then designed to use an adapter to whatever Learning Management System is targeted, which in the case of our prototype is Moodle. This connection is mostly leveraged to get user information to the platform, which allows us to link activities and analytics to a given ID. For the supply of activities, we integrate with activity providers, which are also external agents to the Inven!RA system.

Lastly, we have the authoring tool UI prototype, which is a React web application that uses the REST API of the Node.js backend to gather and manipulate all the necessary data.

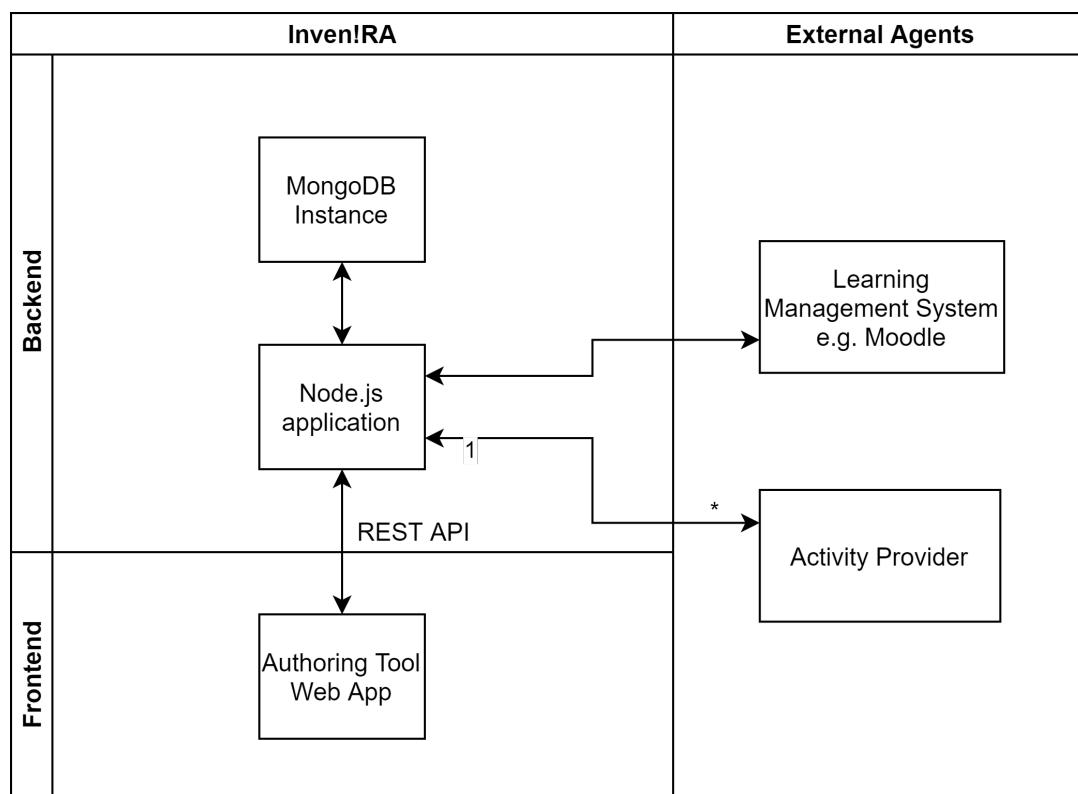


Figure 4.1: High Level Architecture Diagram

4.4 System Architecture

It was clear from the start that we wanted a modular, flexible but robust architecture. The plans to integrate with other ongoing projects while specifying and developing the platform, made it even more of a necessity.

The main focus of the project, as previously stated, was the backend application, and we defined that any visualization, interpretation or implementation particulars could and should only be done on any frontend that connected to a sufficiently flexible backend. That is why the backend was designed to be modular and easy to integrate.

For the frontend web app, the main objective was proving the concept and visualize what is going on the backend in a more natural way.

On [Figure 4.2](#), we have a more detailed look at the architecture through a components diagram.

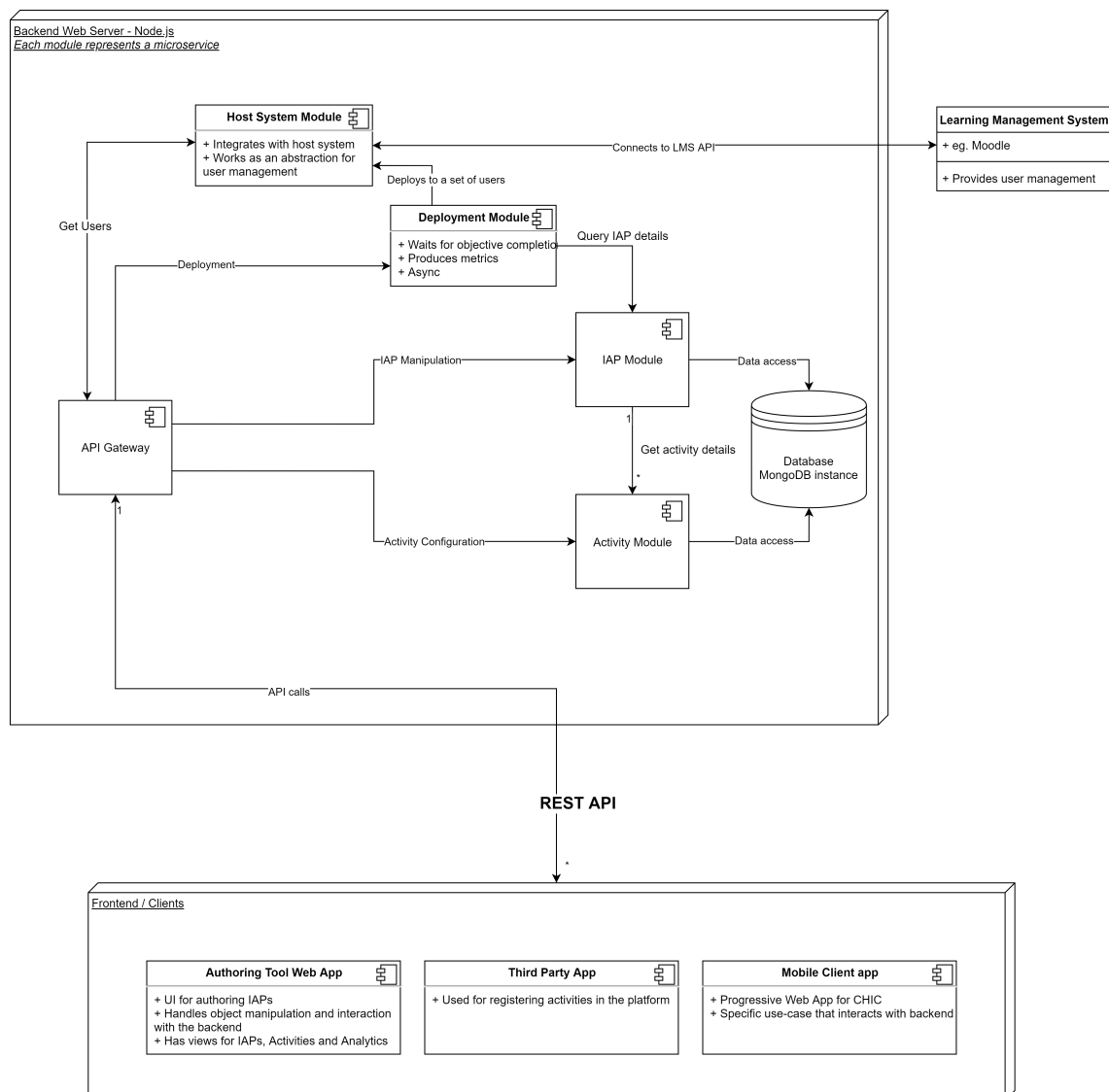


Figure 4.2: Components Diagram

Looking back at BEACONING’s authoring tool, which was based on PHP, namely Laravel, the use of an MVC architecture might have provided a more closely-knit package, but it seriously limits expansion, which led us to not leverage so much the code as we did the concepts that led to the implementation.

In the next subsections we will provide an overview for each of the modules present on Figure 4.2 and their functionality.

4.4.1 Backend

The modules on the backend all represent a microservice, capable of staying up even if another one fails, by relying on async operations and on a lifecycle broker provided by Moleculer.

4.4.1.1 API Gateway Module

This module is the entry point of the backend application. It handles the requests and forwards them to the relevant module.

4.4.1.2 IAP Module

The module responsible for IAP CRUD operations, it provides functions for all required functionality, and validations as needed. It connects to the database through a mixin. This module also connects to the activity module to gather necessary information as needed.

4.4.1.3 Activity Module

The activity module consists of functions for activity manipulation, both CRUD, which will be exposed to third party activity providers, and other configurations needed. It connects to the MongoDB through a mixin.

4.4.1.4 Deployment Module

This module is responsible for providing the necessary functions for deploying an IAP. This can go from gathering info from the IAP module to validation that all necessary data for deployment is present, going all the way to gathering the user information and handling activity deployment triggers. In this module we also have the required functions for analytics.

4.4.1.5 Host System Module

The host system module works as an abstraction for the target LMS. In the scope of this work, we will provide a proof of concept that connects with Moodle for gathering user data. The goal for this module would be to provide hooks for several LMS and abstracting the rest of the application from any LMS specific implementations.

4.4.2 Frontends

While the main focus of the present work is to provide an authoring tool frontend to prove the backend features and experiment, it is important to delineate the compatibility and the objective to incorporate several frontends, which can provide different features. We will address some frontends already developed.

4.4.2.1 Authoring Tool Web App

Our prototype implementation for Inven!RA's authoring tool, in the form of a React Progressive Web App. It provides the most important views for authoring IAPs and visualizing backend data.

4.4.2.2 Third Party App

This module represents any third-party app from an activity provider, which integrates with our system. As a real-life example, we can reference Duarte Cota's code, which we will address in [Chapter 6](#).

4.4.2.3 Mobile Client App

On [Chapter 6](#) we will also address in more detail the previously mentioned CHIC project, which makes use of Inven!RA's backend for a mobile application aimed at tourist experiences.

4.5 Sequence Diagrams

Following the user stories, and after we designed the overall the architecture of the system, we got back in contact with the Inven!RA team to design flow diagrams for the most simple but essential actions, already taking into consideration the envisioned modules.

This process helped us understand the required user interactions on the frontend side and how they would translate to the backend implementation. These flows also helped us in the continuous validation of the solution, being constant artefacts we got back to consult and see if the solution was on track.

4.5.1 Insert activity in a plan

This sequence arises during activity 3.3.1, "Creating and Editing", and details what happens among components in the two subprocesses therein: editing/creating an IAP; and while editing an IAP add activities to it, which originates the need to configure each one.

The Learning Designer initiates this process by editing the IAP, as shown in Figure 3.1. How this actually occurs depends on how each specific frontend is designed and implemented. For instance, there could be a button "add activity" which presents a list of available activities; or all available activities could already be visible from the beginning of the process. Possibly, there could be a voice request, leading to a matching process between the voice recognition outcome and the existing activities. All these, or other, methods of interaction are only frontend related and do not impact the core backend operation.

So, we abstract this by considering as a starting point in this sequence that the Learning Designer requests a list of activities (top sequence) and then drags one of those activities to the IAP (middle sequence). The "dragging" of the activity to an IAP illustrates whatever actual action the specific frontend implements to add an activity to the IAP. This inclusion leads the backend to get the configuration parameters from the database, which include the configuration page endpoint on the third-party provider. The frontend then serves the configuration, for example by embedding the target configuration page, and knows what objects to track based on the configuration json retrieved.

Upon the filling of the configuration form (or other configuration approach), the data is sent to the backend, which will validate it, and upon successful validation, store the required objects in the database.

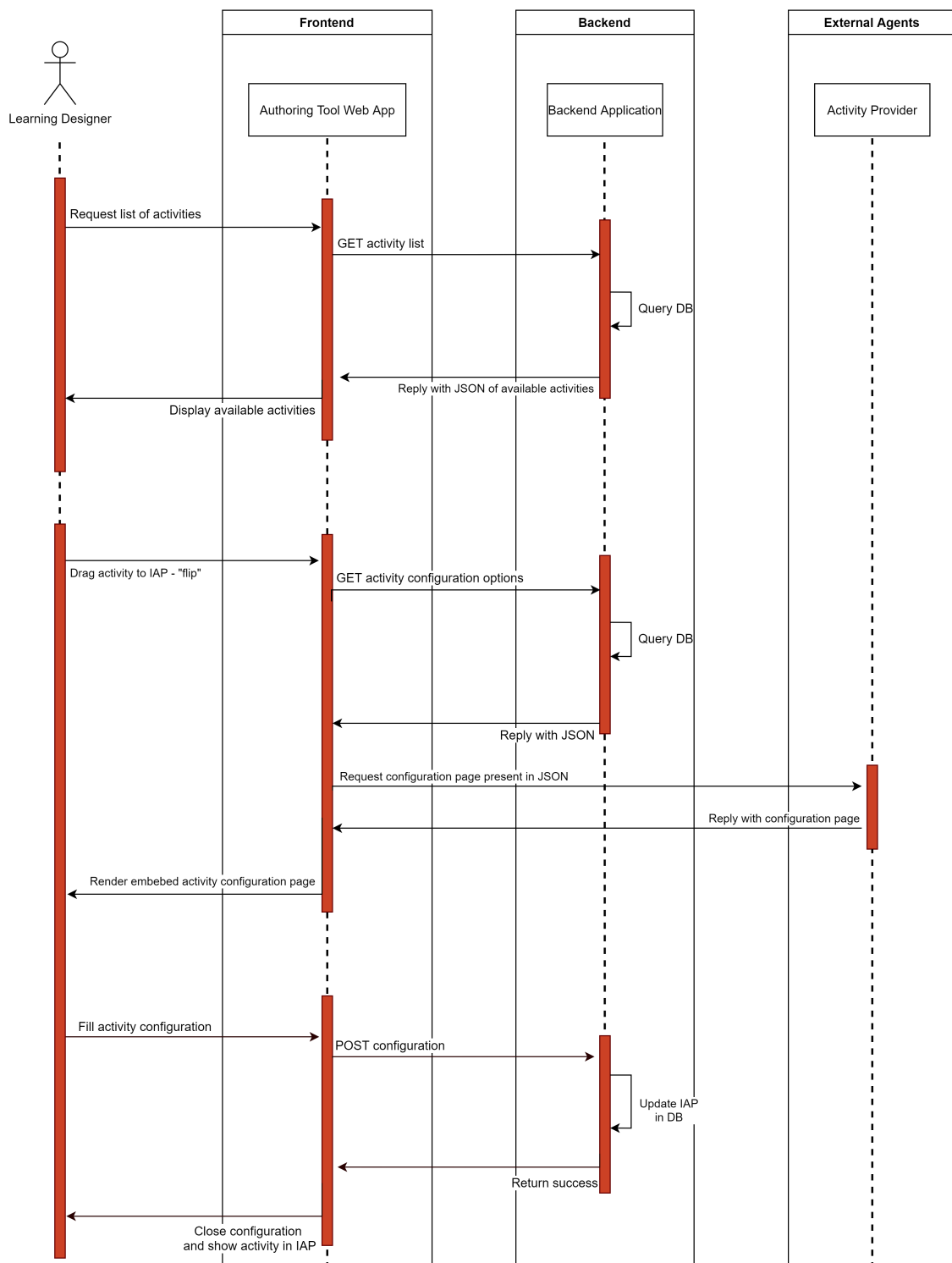


Figure 4.3: Flow Diagram - Insert activity in IAP

4.5.2 Assign pedagogic objective to a plan

The final step when configuring an IAP is to set its learning objectives, as evidenced on activity diagram 3.3.1.

First, let's assume there is some button or other action on the frontend that triggers the start of insertion of learning objectives in the IAP, as seen on the top sequence on Figure 13. This action will trigger a call to the backend, which will analyse all currently inserted activities in the IAP and retrieve the available analytics for them, which were stored on the database when the activity was first added to the platform. When this action is completed, a JSON containing the compiled information will be served to the frontend.

Having the information to display, the frontend will then require the user to name the objective, and pick which activities and which analytics will contribute towards this goal. When the user is satisfied and saves, the objective object will be added to the IAP on the database.

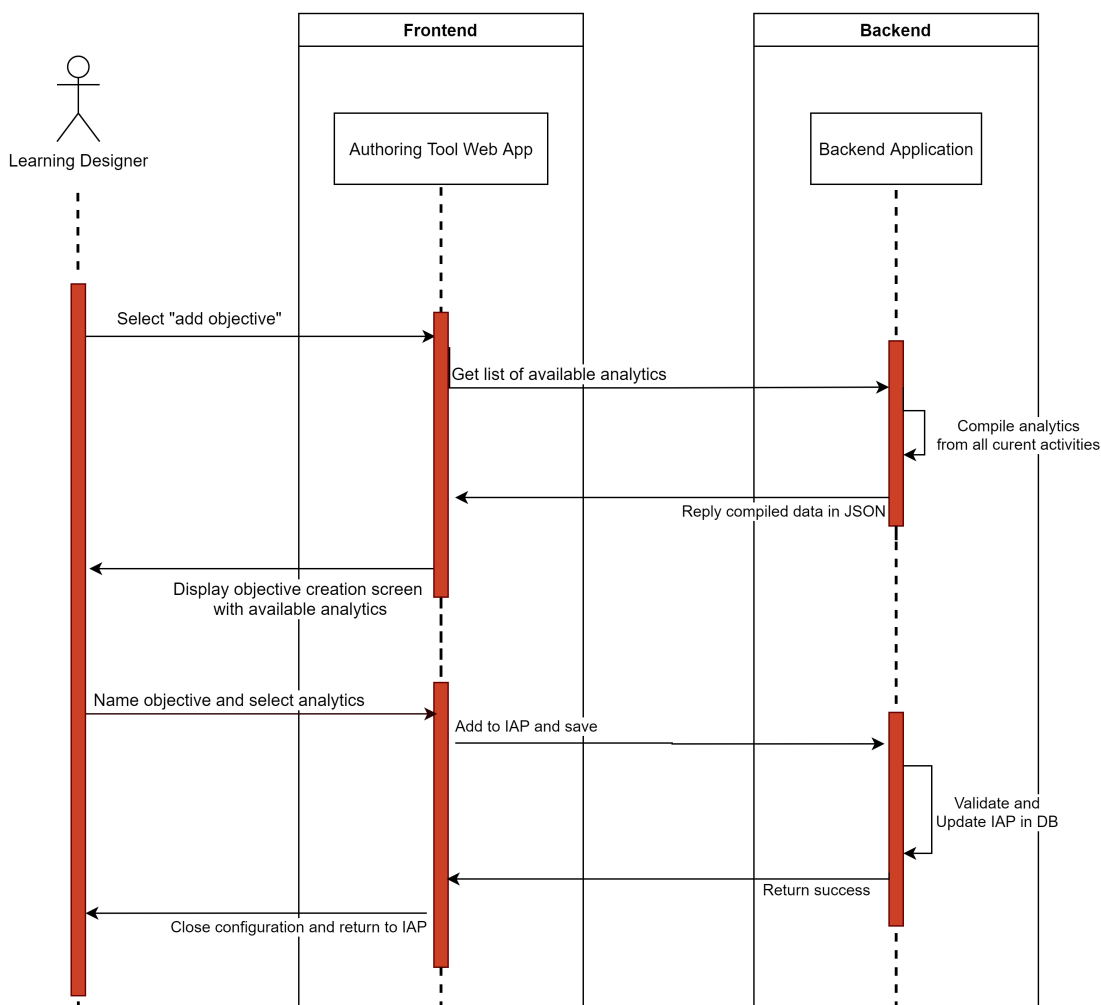


Figure 4.4: Flow Diagram - Assign objective to IAP

4.5.3 Connect activities

This simple sequence comes from the need, as referred on 3.2.1 - US01, to connect the activities that constitute an IAP. These connections are expectably most relevant to the specific implementation of the frontend, as it can chose to evince an oriented flow, or different types of connections

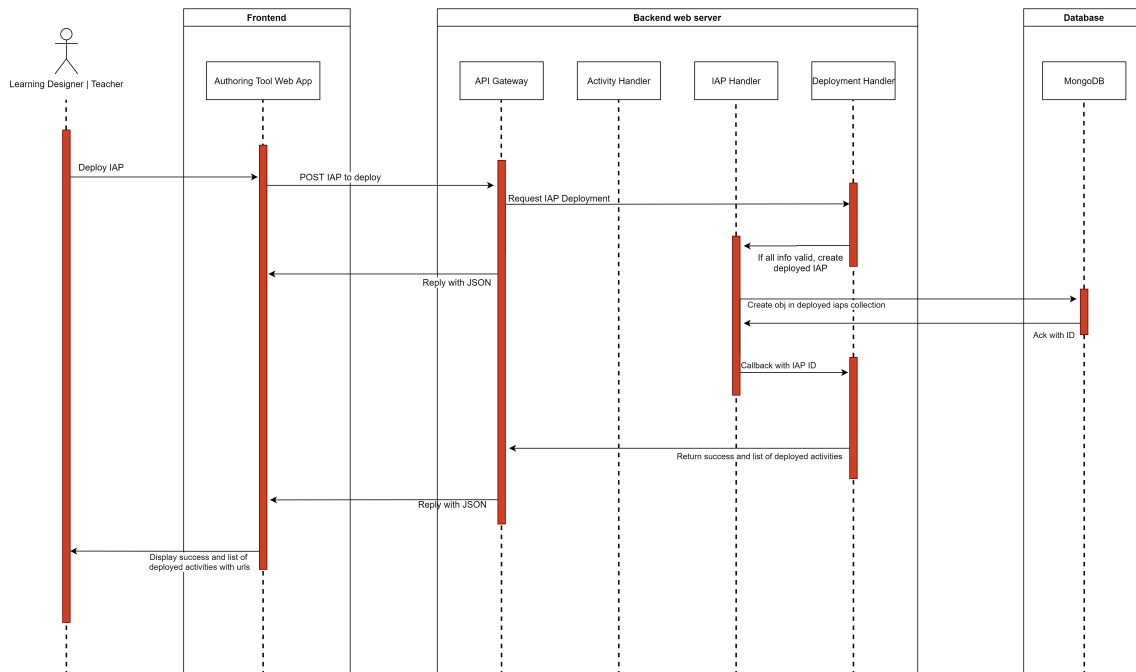


Figure 4.6: Flow Diagram - IAP Deployment

For a diagram of deployments from the activity provider perspective, check Figure 6.15.

4.5.5 Student Activity Deployment

This sequence comes from activity 3.3.3, where a student experiences an IAP. While this action is transparent to the user, Inven!RA's backend will act as middleware to connect the anonymized user to the activity provider.

The sequence starts by the student clicking an URL that was shared by the teacher. This URL was extracted at the end of sequence 4.5.4. For illustration purposes, and as it was the LMS picked for the prototype, we assume the teacher shared the URL using Moodle, and had instructions from the frontend to include the Moodle user id as a parameter to the URL, which is a possible configuration in Moodle.

This URL will trigger the deployment handler on the backend, which will transform the Moodle User ID to Inven!RA's user id, and send the instruction to start the activity to the provider. The provider will then reply with an experience URL, and the backend will reply with a redirect to this activity, which the user can now experience.

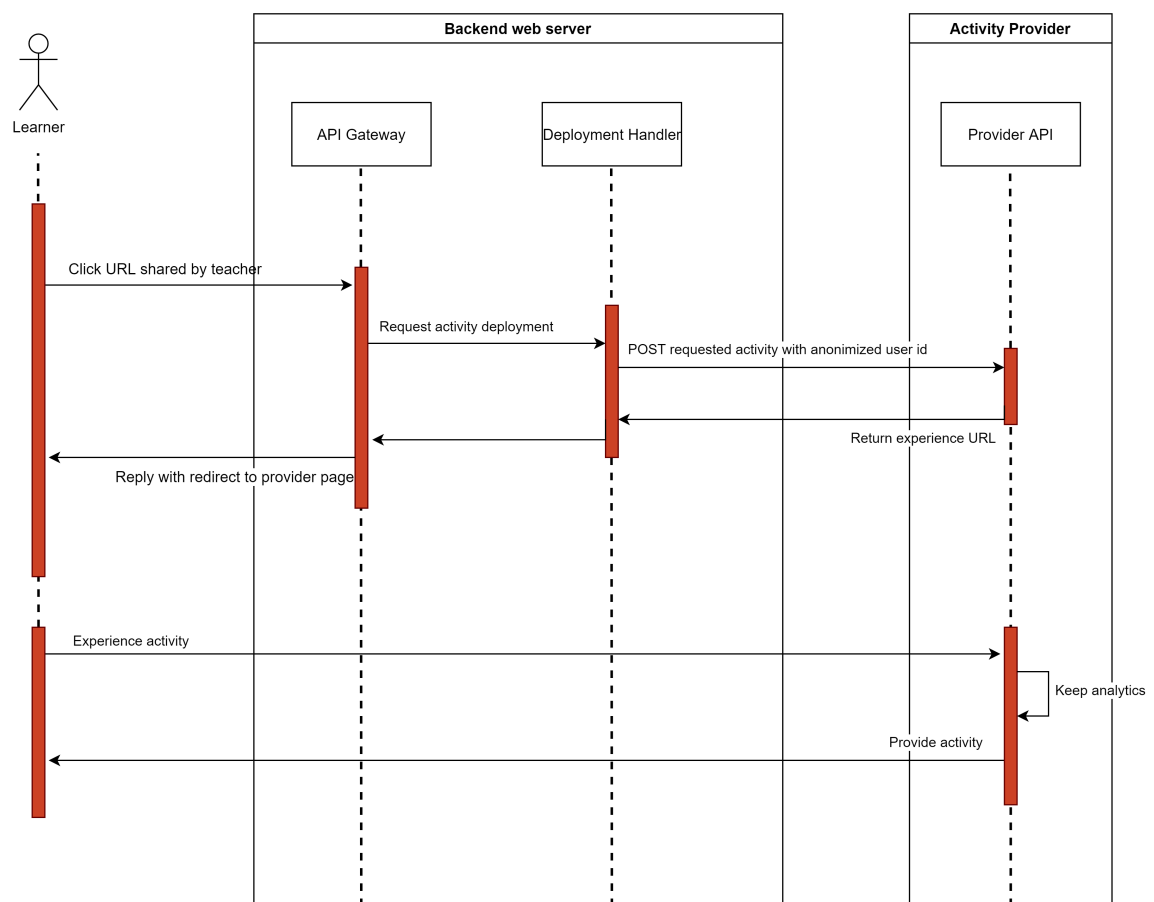


Figure 4.7: Flow Diagram - Student Activity Deployment

Chapter 5

The prototype

In this chapter we will describe and document the most relevant specific implementation details and decisions of the developed prototype and the steps taken to get there, both on the backend and on the frontend. Most importantly, we will document the adopted JSON payloads for the objects and the exposed API calls. We will also provide some screenshots and explanations where relevant, from a technical standpoint. On the frontend side, we will document each implemented view and functionality. The objective of this chapter is to document the software development journey, after architectural decisions were taken.

5.1 Software Development Life Cycle

We want to start by going through some of the decisions taken for the software development life cycle (SDLC), such as repositories, organization, and testing.

As should be clear by now, it was important for us to completely isolate the backend and frontend. It is therefore no surprise that we keep the source code in two separate GIT repositories. Using GIT for code versioning was a no brainer, and we think it can provide value to any individuals that can pick up the project where we left it, with some care being taken into commit messages.

Something that was not initially thought, but ended up being a significant part of the work, was dev ops. As we were continuously integrating with other teams, the need to have the services exposed online was quick to surface, and we took the time to make sure everything is easily configured and steps are documented on how to do it – which we detail in Annex A. For testing and integration with the other teams, we used a Linux Server Virtual Private Server (VPS), with an NGINX server running multiple websites. The backend web services were exposed on NGINX by making use of the reverse proxy function, and running the two instances with Node.js on different ports.

For the database, since we are using MongoDB, we made use of the ATLAS free tier, to run an instance of the database. Since the colleagues on the CHIC project had slightly different requirements and very different data, the server is actually running 2 concurrent instances of Inven!RA,

each connected to a separate MongoDB instance.

For the frontend testing, the same VPS was used, by exposing a compiled version of the React application through NGINX.

All these configurations, in addition with obvious local testing, were improved throughout the development and made easier by the use of .env files for runtime configuration, both on the backend and frontend. We have created easy to follow instructions, and default fallbacks, to easily configure these parameters.

Development was mostly done using Visual Studio Code, and the terminal of choice was Cmdr.

For testing the web services, we made use of the Insomnia Rest Client, where we created sample requests for each operation and used as unit tests.

5.2 Backend

The decision to use a microservices approach to the backend helped us streamline the architecture and provide a simple and effective way to isolate logic for each component, as well as encourage the development of new components in the future. By using Moleculer, a specific format was followed for file naming and structure, as illustrated on Figure 5.1. As can be seen, each microservice is stored on the “services” folder, and identified with “.service.js” in its name. This allows Moleculer to identify these pieces of code as a microservice, enabling their usage.

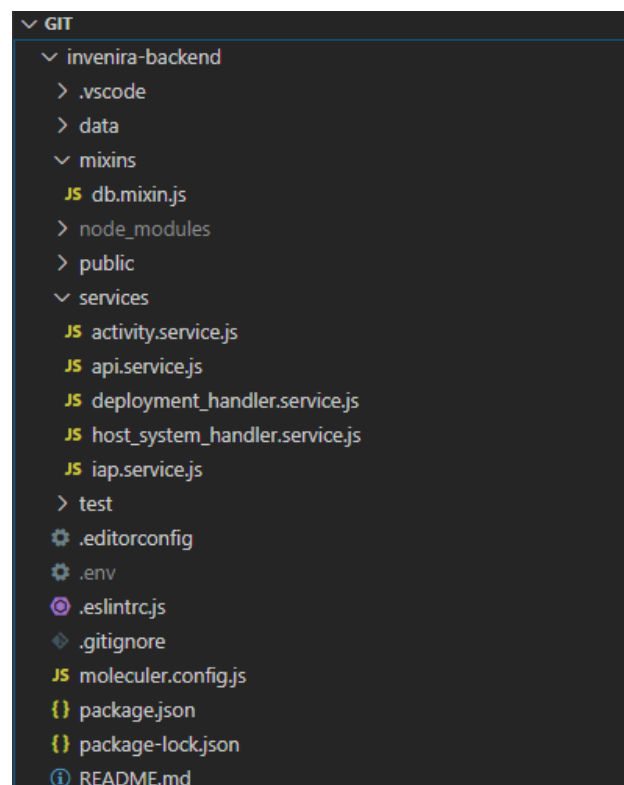


Figure 5.1: Code organization on the backend repository

Each microservice pretty much grew from its related object in the database and is responsible to contain all necessary functions to interact with it.

5.2.1 Services

Going back to Section 4.4.1, we will now address what each microservice specifically does in terms of functionality and how it translates to the end product. By standard, each module has CRUD methods for object manipulation, provided by *Moleculer* and extended as needed, as well as listing methods where it is useful for visualization purposes.

Each microservice is structured by 4 main exports: the name, the used mixins, the settings and actions.

The mixins are used for a single database mixin, which received the MongoDB collection name from each microservice, as the rest of the connection is shared.

The most relevant settings are the REST endpoint and the fields to be used by the built-in entity validator. This entity validator lets us set mandatory fields and their types, which makes the CRUD operations more robust and less prone to data corruption.

The actions are the service-specific functions, which we address as needed in the next subsections. Only the major functionalities and most relevant details will be approached. For a full list of the available operations, we have produced a REST Specification document, available in Appendix B.

5.2.1.1 IAP Handler

The IAP handler microservice overrides the default methods to get, create, and list IAPs. In terms of validation, at this stage only the name is mandatory when creating an IAP.

The JSON Schema for an IAP is constituted by an ID, a name, an array of properties, an array of nodes and an array of edges. A sample can be seen on Figure 5.2.

This payload allows us to represent the graph of an IAP, and identify important information for each node, which represents an activity. As a rule of thumb for each of our payloads, we included an array of properties, which gives us the flexibility to expand at any given time with no need to change the schema.

When calling the list operation, an array with all available IAPs in the database will be returned.

For the get operation, the service expects an ID as URL parameter, in the format `host:port/api_endpoint/iap/#IAPID`.

```

1 {
2   "_id": "5eee571ea691f8225432699a",
3   "name": "Porto Exploration",
4   "properties": [
5     {
6       "description": "This is an IAP with LBS activities"
7     }
8   ],
9   "nodes": [
10    {
11      "node_id": "act-1",
12      "act_id": "5eb0b6100e0ac238c4e8e623",
13      "name": "One",
14      "properties": [
15        {
16          "pos_x": 150,
17          "pos_y": 150
18        }
19      ]
20    },
21    {
22      "node_id": "act-2",
23      "act_id": "5eb0b5110e0ac238c4e8e621",
24      "name": "Two",
25      "properties": [
26        {
27          "pos_x": 400,
28          "pos_y": 150
29        }
30      ]
31    },
32    {
33      "node_id": "act-3",
34      "act_id": "5eb0b5f50e0ac238c4e8e622",
35      "name": "Three",
36      "properties": [
37        {
38          "pos_x": 400,
39          "pos_y": 450
40        }
41      ]
42    }
43  ],
44  "edges": [
45    {
46      "source": "act-1",
47      "target": "act-2",
48      "label": ""
49    }
50  ]
51 }

```

Figure 5.2: IAP Sample payload

The create operation expects an IAP, where only the name is mandatory. This operation will insert the object in the database and return its ID.

5.2.1.2 Activity Handler

The activity handler microservice is responsible for activity operations. It is, therefore, of major importance for the third party activity providers, who must use it to configure activities in the system, by calling the create operation with a compatible payload, which will return an ID upon successful insertion in the database. A sample request and response can be found on Figure 5.3.

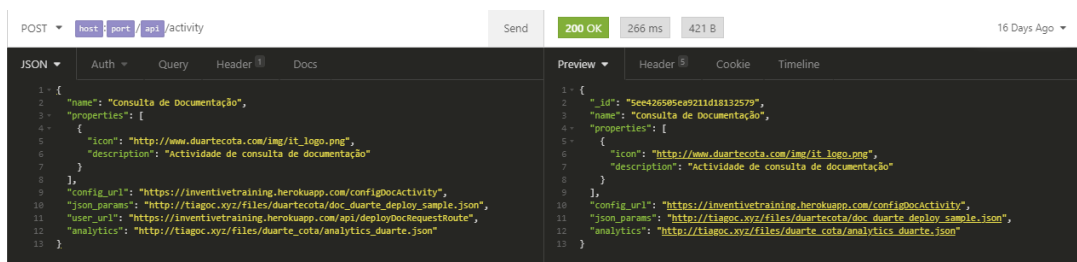


Figure 5.3: Create activity request and response

For the activity object, all fields but the properties are mandatory, as they all have implications in the functionality.

Other than the activity type name, the functionality for each field is as follows:

- **properties:** A flexible array of properties. The icon and description tags can be used to enhance UI experience in the prototype frontend, but have no backend relevance
- **config_url:** The url for the configuration page of the activity. This will be loaded when the activity is dragged to an IAP by a learning designer
- **json_params:** An url to a json file with a list of parameters to extract from the configuration page
- **user_url:** The url that will be used by the learner after deployment
- **analytics:** An url to a json file containing the analytics for the activity

The list action will provide a list of all configured activities and is used to load them on the main IAP editing page. Filters can be put in place, but at this point we are getting all available activities.

All CRUD operations are also available on the service, with PUT providing update functionality, and the ID being the only required field to GET or DELETE.

5.2.1.3 Deployment Handler

The deployment handler microservice is the most logic-heavy microservice currently in place. It expands the IAP concept from the IAP Handler, but adds an objectives array to the schema and additional validations, to make sure all activities were fully configured before the IAP is deployed.

The two most relevant actions are as follows:

- **Get Activities for IAP:** This action returns a list of deployment URLs for each activity present in a deployed IAP. These URLs can then be used by the teacher to provide the learners with the entry point for their activities. A sample call can be seen on Figure 5.4.
- **Initiate Activity:** This action gets 2 parameters as query parameters in the URL: the activity ID and the User ID. With this information, the service then queries the host system handler

to get the Moodle ID for the user, and initializes the activity to start producing analytics. The user is then redirected to the user URL of the activity, with all necessary configuration being posted to the provider in a JSON file.

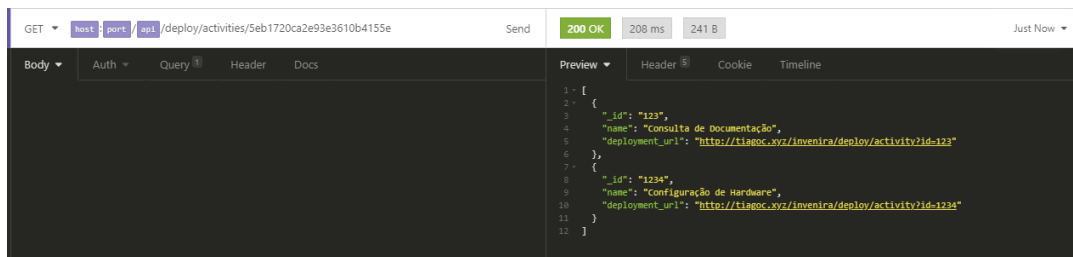


Figure 5.4: Get activities for deployed IAP

5.2.1.4 Host System Handler

This microservice is, at this point, just a user abstraction that connects with Moodle. The methods available are GET, which gets a list of available users on Moodle, and the internal *init_moodle* method initializes the API connection to Moodle.

The GET operation returns a list of objects with 3 fields, the internal user ID, the Moodle ID, and the groups to which the user belongs.

This service was not fully developed, as it wasn't a main goal of this iteration of Inven!RA to provide an extensive integration with the LMS, so it is expected that more operations would be needed and the payloads may need to change.

5.2.2 Running the backend

To start the backend, some configurations should be done first, notably the creation of the *.env* file for environment variables, as well as running *"npm install"* to install the required dependencies.

On Figure A.1 we can see the required variables, represented in this case by the default values.

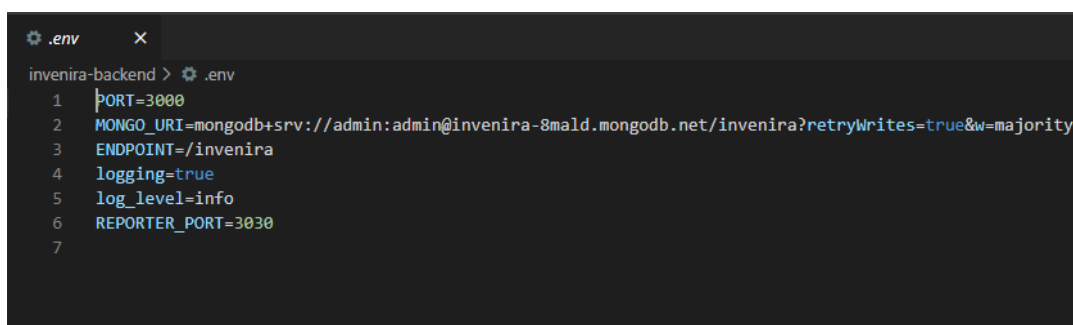


Figure 5.5: Environment variables for the backend

When everything is set up, the application should be launched by running *"npm run mol"*, which will start up node and run the Moleculer service broker, illustrated on Figure 5.6.

```

C:\Users\user> npm run mol
invenira-backend@1.0.0 mol C:\GIT\invenira-backend
> moleculer-runner --repl --hot services/**/*.service.js -e

[2020-06-29T02:09:33.106Z] INFO desktop-od2mc3c-792032/BROKER: Moleculer v0.14.4 is starting...
[2020-06-29T02:09:33.109Z] INFO desktop-od2mc3c-792032/BROKER: Namespace: <not defined>
[2020-06-29T02:09:33.110Z] INFO desktop-od2mc3c-792032/BROKER: Node ID: desktop-od2mc3c-792032
[2020-06-29T02:09:33.111Z] INFO desktop-od2mc3c-792032/METRICS: Metrics: Enabled
[2020-06-29T02:09:33.113Z] INFO desktop-od2mc3c-792032/METRICS: Metric reporter: PrometheusReporter
[2020-06-29T02:09:33.148Z] INFO desktop-od2mc3c-792032/REGISTRY: Strategy: RoundRobinStrategy
[2020-06-29T02:09:33.152Z] INFO desktop-od2mc3c-792032/BROKER: Serializer: JSONSerializer
[2020-06-29T02:09:33.162Z] INFO desktop-od2mc3c-792032/TRACER: Tracing: Enabled
[2020-06-29T02:09:33.163Z] INFO desktop-od2mc3c-792032/TRACE: Tracing exporter: ConsoleTraceExporter
[2020-06-29T02:09:33.165Z] INFO desktop-od2mc3c-792032/BROKER: Registered 14 internal middleware(s).
[2020-06-29T02:09:33.755Z] INFO desktop-od2mc3c-792032/API: API Gateway server created.
[2020-06-29T02:09:33.757Z] INFO desktop-od2mc3c-792032/API: Register route to '/invenira'
[2020-06-29T02:09:33.786Z] INFO desktop-od2mc3c-792032/API: Generate aliases for '/invenira' route...
[2020-06-29T02:09:33.787Z] INFO desktop-od2mc3c-792032/API:
[2020-06-29T02:09:33.130Z] INFO desktop-od2mc3c-792032/REGISTRY: '$node' service is registered.
[2020-06-29T02:09:33.133Z] INFO desktop-od2mc3c-792032/REGISTRY: 'host_system_handler' service is registered.
[2020-06-29T02:09:33.135Z] INFO desktop-od2mc3c-792032/REGISTRY: '$node' started.
[2020-06-29T02:09:33.135Z] INFO desktop-od2mc3c-792032/HOST_SYSTEM_HANDLER: Service 'host_system_handler' started.
[2020-06-29T02:09:33.137Z] INFO desktop-od2mc3c-792032/METRICS: Prometheus metric reporter listening on http://0.0.0.0:3030/metrics address.
[2020-06-29T02:09:33.139Z] INFO desktop-od2mc3c-792032/API: API Gateway listening on http://localhost:3000
[2020-06-29T02:09:33.141Z] INFO desktop-od2mc3c-792032/REGISTRY: 'api' service is registered.
[2020-06-29T02:09:33.143Z] INFO desktop-od2mc3c-792032/API: Service 'api' started.
(node:792032) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new
Server Discovery and Monitoring engine, pass option { unifiedTopology: true } to the MongoClient constructor.
[2020-06-29T02:09:33.642Z] INFO desktop-od2mc3c-792032/API: Generate aliases for '/invenira' route...
[2020-06-29T02:09:33.644Z] INFO desktop-od2mc3c-792032/API: GET /invenira/users => host_system_handler.list
[2020-06-29T02:09:33.646Z] INFO desktop-od2mc3c-792032/API: GET /invenira/api/list-aliases => api.listAliases
[2020-06-29T02:09:33.700Z] INFO desktop-od2mc3c-792032/ACTIVITY: MongoDB adapter has connected successfully.
[2020-06-29T02:09:33.704Z] INFO desktop-od2mc3c-792032/REGISTRY: 'activity' service is registered.
[2020-06-29T02:09:33.705Z] INFO desktop-od2mc3c-792032/ACTIVITY: Service 'activity' started.
[2020-06-29T02:09:33.707Z] INFO desktop-od2mc3c-792032/IAP: MongoDB adapter has connected successfully.
[2020-06-29T02:09:33.710Z] INFO desktop-od2mc3c-792032/DEPLOYMENTHANDLER: MongoDB adapter has connected successfully.
[2020-06-29T02:09:33.715Z] INFO desktop-od2mc3c-792032/REGISTRY: 'iap' service is registered.
[2020-06-29T02:09:33.727Z] INFO desktop-od2mc3c-792032/REGISTRY: 'deploymenthandler' service is registered.
[2020-06-29T02:09:33.729Z] INFO desktop-od2mc3c-792032/IAP: Service 'iap' started.
[2020-06-29T02:09:33.730Z] INFO desktop-od2mc3c-792032/DEPLOYMENTHANDLER: Service 'deploymenthandler' started.
[2020-06-29T02:09:33.733Z] INFO desktop-od2mc3c-792032/BROKER: Hot-reload is ACTIVE.
[2020-06-29T02:09:33.736Z] INFO desktop-od2mc3c-792032/BROKER: ServiceBroker with 6 service(s) is started successfully in 613ms.
mol $ [2020-06-29T02:09:35.318Z] INFO desktop-od2mc3c-792032/API: Generate aliases for '/invenira' route...
[2020-06-29T02:09:35.319Z] INFO desktop-od2mc3c-792032/API: GET /invenira/users => host_system_handler.list
[2020-06-29T02:09:35.319Z] INFO desktop-od2mc3c-792032/API: GET /invenira/api/list-aliases => api.listAliases
[2020-06-29T02:09:35.319Z] INFO desktop-od2mc3c-792032/API: GET /invenira/activity => activity.list
[2020-06-29T02:09:35.319Z] INFO desktop-od2mc3c-792032/API: POST /invenira/activity => activity.create
[2020-06-29T02:09:35.319Z] INFO desktop-od2mc3c-792032/API: GET /invenira/activity/:id => activity.get
[2020-06-29T02:09:35.319Z] INFO desktop-od2mc3c-792032/API: PUT /invenira/activity/:id => activity.update
[2020-06-29T02:09:35.321Z] INFO desktop-od2mc3c-792032/API: DELETE /invenira/activity/:id => activity.remove
[2020-06-29T02:09:35.323Z] INFO desktop-od2mc3c-792032/API: GET /invenira/iap => iap.list
[2020-06-29T02:09:35.323Z] INFO desktop-od2mc3c-792032/API: POST /invenira/iap => iap.create
[2020-06-29T02:09:35.327Z] INFO desktop-od2mc3c-792032/API: GET /invenira/iap/:id => iap.get
[2020-06-29T02:09:35.329Z] INFO desktop-od2mc3c-792032/API: PUT /invenira/iap/:id => iap.update
[2020-06-29T02:09:35.330Z] INFO desktop-od2mc3c-792032/API: DELETE /invenira/iap/:id => iap.remove
[2020-06-29T02:09:35.332Z] INFO desktop-od2mc3c-792032/API: GET /invenira/deploy => deploymenthandler.list
[2020-06-29T02:09:35.333Z] INFO desktop-od2mc3c-792032/API: POST /invenira/deploy => deploymenthandler.create
[2020-06-29T02:09:35.335Z] INFO desktop-od2mc3c-792032/API: GET /invenira/deploy/:id => deploymenthandler.get
[2020-06-29T02:09:35.337Z] INFO desktop-od2mc3c-792032/API: PUT /invenira/deploy/:id => deploymenthandler.update
[2020-06-29T02:09:35.338Z] INFO desktop-od2mc3c-792032/API: DELETE /invenira/deploy/:id => deploymenthandler.remove
[2020-06-29T02:09:35.339Z] INFO desktop-od2mc3c-792032/API: GET /invenira/deploy/activities/:id => deploymenthandler.getActivitiesForIAP
[2020-06-29T02:09:35.341Z] INFO desktop-od2mc3c-792032/API: GET /invenira/deploy/activity => deploymenthandler.initiateActivity
[2020-06-29T02:09:35.342Z] INFO desktop-od2mc3c-792032/API: GET /invenira/deploy/iap/:id => deploymenthandler.getDeployedIAP
[2020-06-29T02:09:35.343Z] INFO desktop-od2mc3c-792032/API: GET /invenira/deploy/list => deploymenthandler.listDeployedIAPs

```

Figure 5.6: Backend log output

5.3 Frontend

The frontend web application, developed using React, aims to provide a visual aid in understanding the project, as well as proving the backend functionality and even improving it by using it as a real end user would.

For the purpose of this work, we have implemented the following screens and functionalities:

- Navigation
 - Top navigation bar provides quick shortcuts to visualize available IAPs, deployed IAPs and available activities, as well as creating a new IAP
- Home Screen
 - Lists all available IAPs and some of their relevant details, and provides shortcuts to edit or deploy them

- Deployed IAPs
 - Lists all deployed IAPs and relevant details, and provides shortcuts to visualize (read-only) and get URLs for each activity
- Activities
 - Lists all available activities and relevant details. Provides links to check analytics and access configuration payloads
- Analytics
 - Displays available analytics for a given activity
- Create/Edit IAP
 - Provides a user interface to create or edit an IAP

5.3.1 Implementation

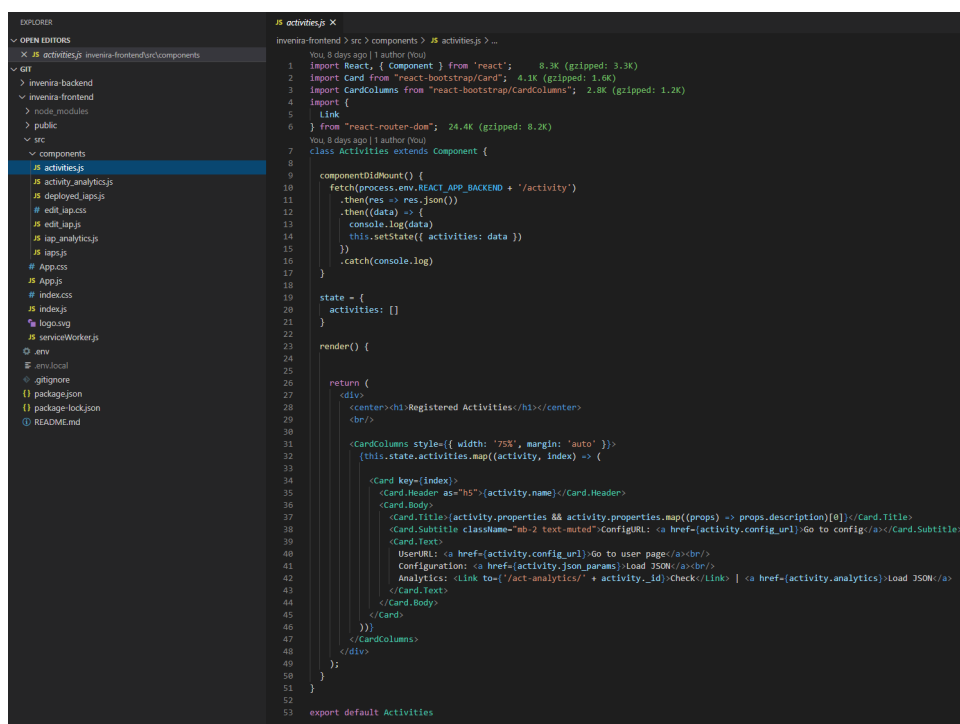


Figure 5.7: Frontend Code Organization

5.3.1.1 Visual Prototyping

We started the fronted implementation by doing a proof of concept (PoC) using some of the most challenging aspects we identified, in the form of real time graph manipulation and visualization and drag and drop support. The first static prototype can be seen on Figure 5.8.

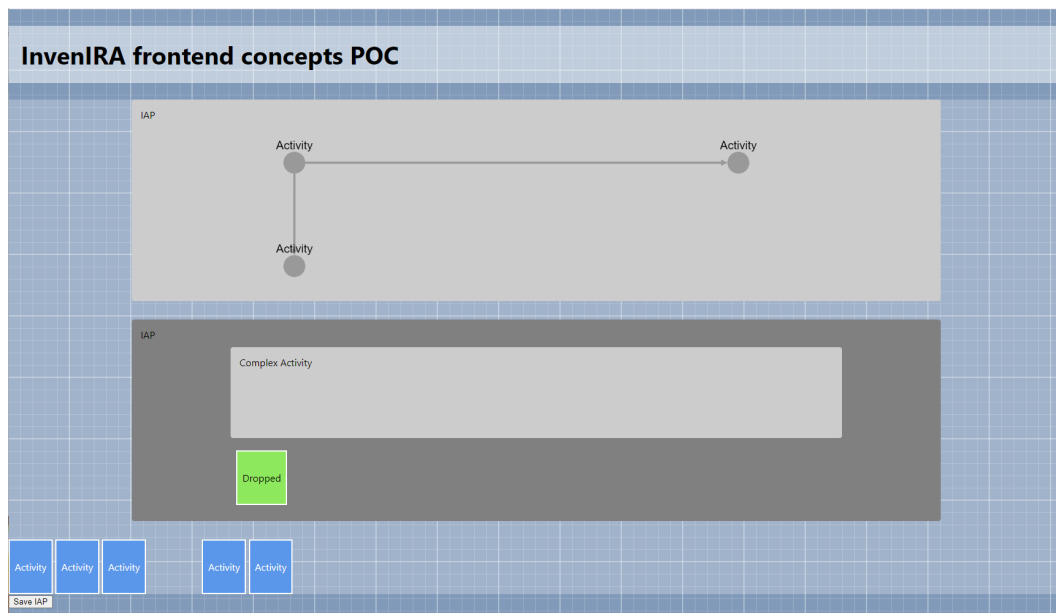


Figure 5.8: Original PoC for the frontend requirements

We used this experience to experiment with frameworks for those purposes, ending up with the aforementioned Cytoscape library, and also using Interact.js for drag and drop support.

5.3.1.2 Frontend Prototype

Moving on to the final React implementation with live data, we organized the code in the usual and recommended React components way.

Each component is mostly comprised of three major parts:

- `componentDidMount` logic – This is the code executed on object load, and is where we load the data from the backend, by making usage of JavaScript's `fetch` function
- `state` – The state is where objects are kept in memory. We use this to store any data being currently used in the view
- `render` – This is where the html objects are declared, in this case mostly React Bootstrap components

While the frontend was not the major focus of the development, it brought some of the biggest challenges, such as graph manipulation, which makes the `edit_iap` component probably the most logic-heavy file in the whole project. In this component, we import and configure the Cytoscape library, expanding it with two native extensions: *cytoscape-cxtmenu*, for the context menu on each node; and *cytoscape-edgehandles*, for connecting nodes and live editing.

Despite not being a requirement, it was always important to the team to produce something that could be easily picked up and expanded in the future, and considerations such as mobile readiness were taken. We took a special keen to making the views compatible with tablet screen,

as we think these would provide ideal tools for learning designers. A screenshot of the tablet view of the editing IAP page can be seen on Figure 5.9. Some testing and precautions were also taken to ensure all interactions are possible on touchscreens, such as long pressing an activity to show its context menu.

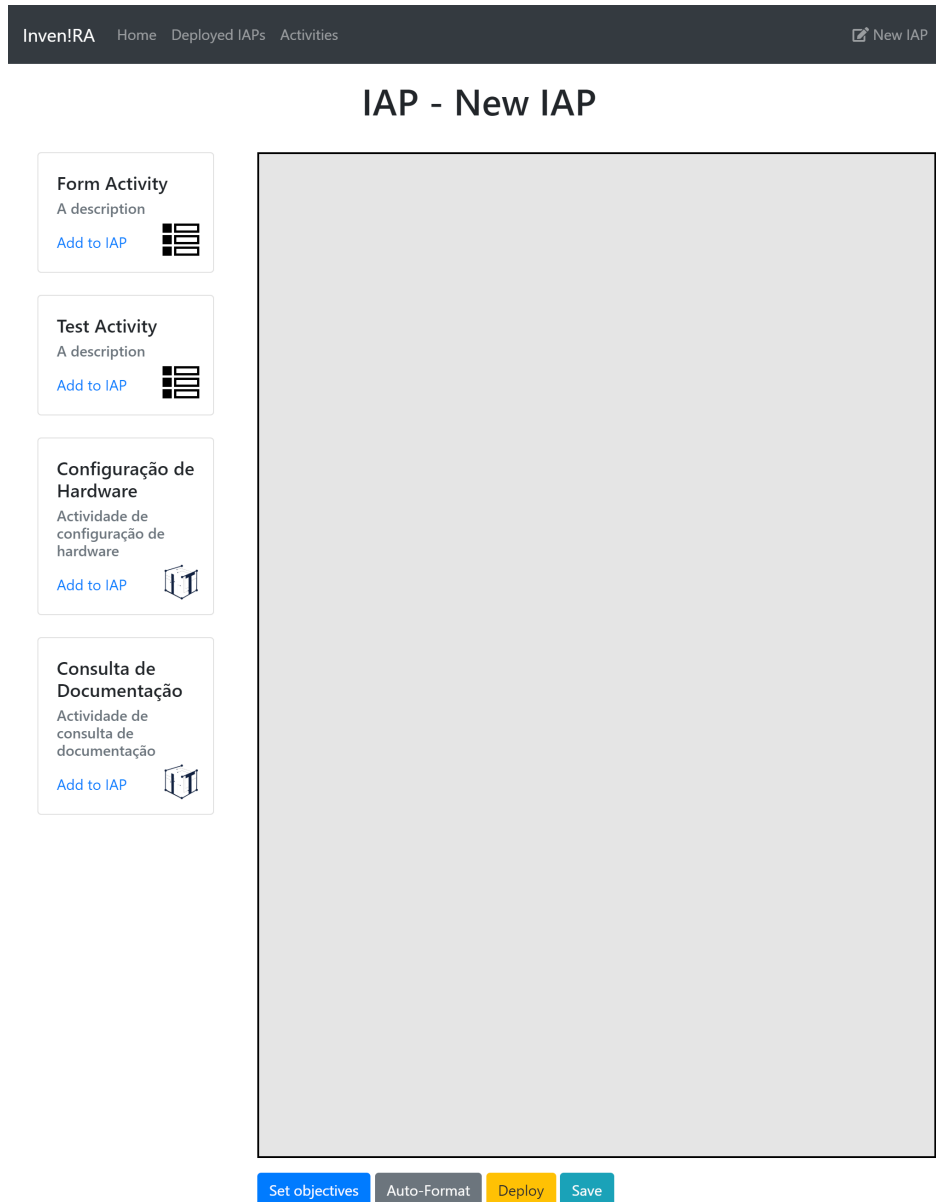


Figure 5.9: Tablet view of the authoring tool

On Chapter 6 we will go through all the views, in the form of formal testing of the platform.

5.3.1.3 Running and deploying the frontend

Running the frontend application goes through the usual flow of JavaScript software, so you should start by cloning the repository and running “*npm install*” to install all dependencies.

After that, you should set the `.env` environment file, which let you set the port where you which the application to run, as well as the URL for the backend, as show by the default configuration on Figure 5.10.

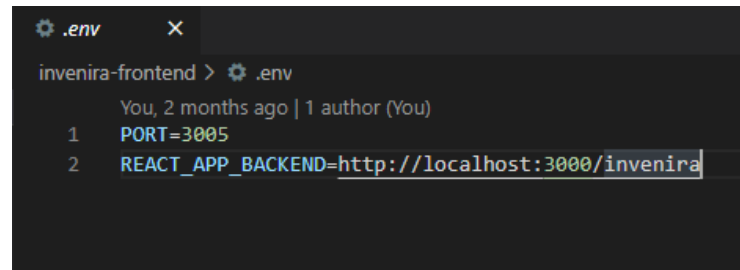


Figure 5.10: Frontend environment file

For running the application locally, you can now run “*npm run start*”, which will launch it on the configured port. If, however, you want to deploy it on a server, you should run “*npm run build*”, which will create all necessary artefacts. More instructions can be found on Appendix A.

Chapter 6

Validation and testing

In the following chapter we will expose and perform a critical analysis on the testing and validation of the prototype. We begin by addressing the unit testing done throughout the development of the prototype, then describing frontend validation of the backend, followed by integration testing, which will take a close look at specific use cases by other researchers that relied on our prototype for their own work, and their experiences with it, as well as their feedback and the feedback of the Inven!RA team.

6.1 Unit Testing

Unit testing was done throughout the development of the project to test the backend functionality at an atomic level. A unit test was created for each of the possible operations on the backend, and ran often, as payloads and logic evolved. This process was also very important in the integration with other teams and in the specification of the JSON payloads.

We used the Insomnia REST Client for these tests, and they are available in the backend repository.

Steps were taken to remove any hassle from the execution, such as setting dynamic configurable URLs for the services, with the file providing by default a Localhost and a Test environment, which points to the author's private VPS used for testing.

A summary of the available unit tests and current execution result can be found on [Table 6.1](#).

6.2 Visual testing

In this section we will illustrate the usage of the authoring tool prototype, by going through and describing all views and their functionality. By using the frontend prototype and validating that all required API calls are happening, we can consider this a formal test of the backend, in addition to a proof of concept and walkthrough for the frontend. In the next subsections we will go through every frontend view and describe what is happening on the backend.

	Test Name	Relative Path	Method	Request Payload	Expected Response	Status on latest build
1	List IAPs	/iap	GET	N.A.	List of available IAPs	Pass
2	Get IAP by ID	/iap/:id	GET	N.A.	IAP Object	Pass
3	Create IAP	/iap	POST	IAP JSON	IAP with created ID	Pass
4	Update IAP	/iap/:id	PUT	IAP JSON	IAP or descriptive Error	Pass
5	Delete IAP	/iap/:id	DELETE	N.A.	Success or 404	Pass
6	List Activities	/activity	GET	N.A.	List of available activities	Pass
7	Get Activity by ID	/activity/:id	GET	N.A.	Activity Object	Pass
8	Create Activity	/activity	POST	Activity JSON	Activity with created ID	Pass
9	Update Activity	/activity/:id	PUT	Activity JSON	Activity or descriptive Error	Pass
10	Delete Activity	/activity/:id	DELETE	N.A.	Success or 404	Pass
11	List Users	/users	GET	N.A.	List of available users	Pass
12	List Deployed IAPs	/deploy/list	GET	N.A.	List of deployed IAPs	Pass
13	List Deployed Activities for IAP	/deploy/activities/:id	GET	N.A.	List of activities in deployed IAP	Pass
14	Trigger Deployed Activity	/deploy/activity	GET	Query Parameters: id and username	Redirect to deployment URL	Pass

Table 6.1: Unit Tests

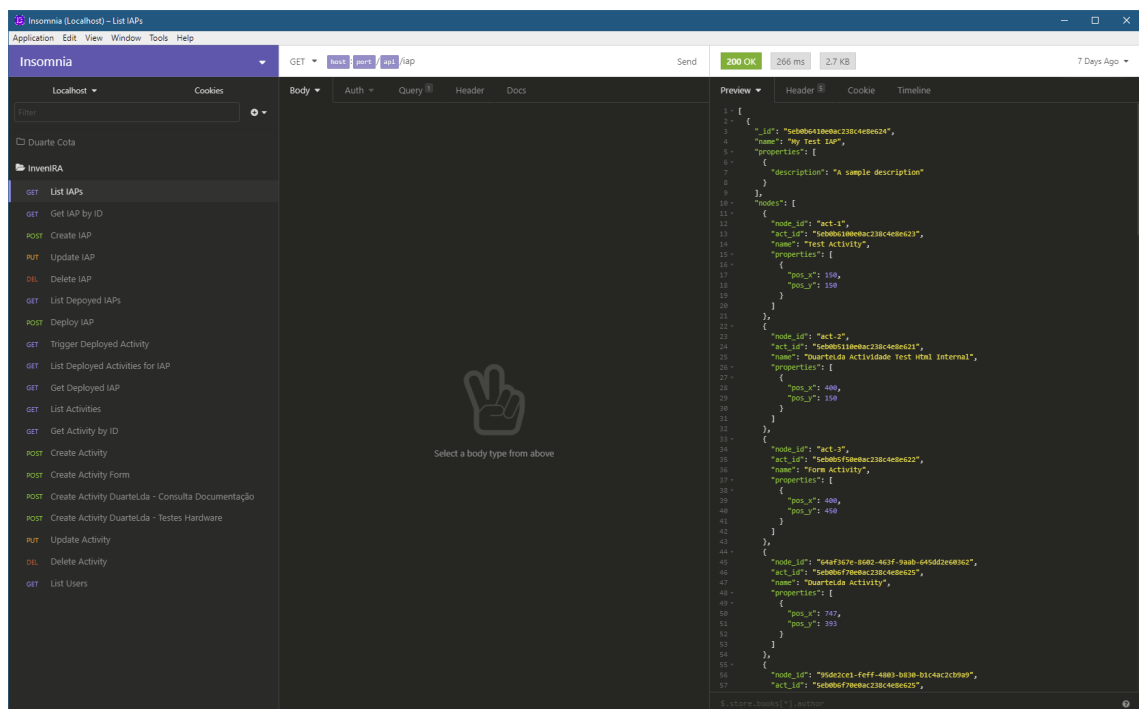


Figure 6.1: Unit test suite in Insomnia

6.2.1 Navigation and Homepage

In the authoring tool homepage, a list of available IAPs is gathered from the IAP List backend functionality and is shown. You can choose to edit them, which will route you to the editing page and load the previously added information.

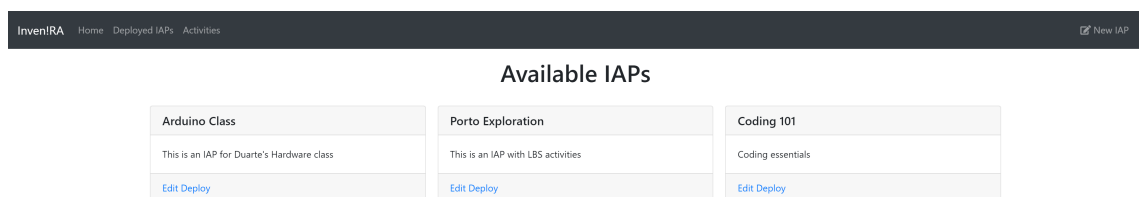


Figure 6.2: Frontend Homepage

6.2.2 Deployed IAPs view

This view differs from the homepage in the fact that it shows already deployed IAPs, which are running and not editable. From this screen, you can consult the deployment URLs for each activity in a given IAP, as well as access the analytics page for that IAP.

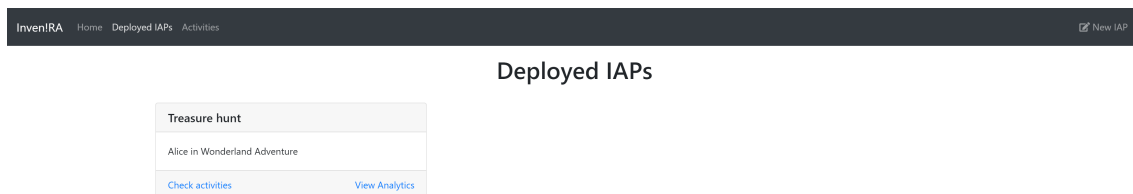


Figure 6.3: Deployed IAPs view

When the user clicks on “Check Activities”, available in each deployed IAP object, there is a call to the backend to load the deployment URLs for each activity configured in that IAP. A table is displayed to the user with this information and user instructions, as seen on Figure 28.

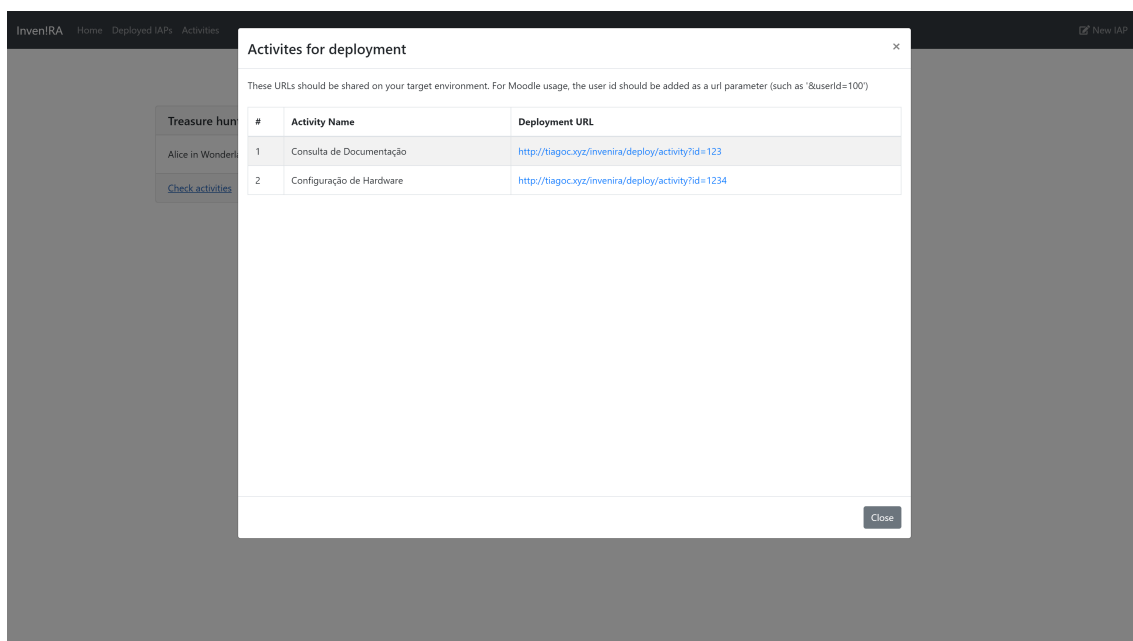


Figure 6.4: Deployed IAP Activities view

6.2.3 Registered activities view

On this page we display a list of all registered activities in this instance of Inven!RA. Useful information, such as configuration payloads and a dedicated analytics page, are available as hyperlinks.

Registered Activities		
Configuração de Hardware Actividade de configuração de hardware ConfigURL: Go to config UserURL: Go to user page Configuration: Load JSON Analytics: Check Load JSON	Consulta de Documentação Actividade de consulta de documentação ConfigURL: Go to config UserURL: Go to user page Configuration: Load JSON Analytics: Check Load JSON	Form Activity An online form ConfigURL: Go to config UserURL: Go to user page Configuration: Load JSON Analytics: Check Load JSON

Figure 6.5: Registered Activities view

By selecting the analytics hyperlink, the backend triggers a call to the providers to query the relevant data, and returns it to the frontend, which interprets it and shows the information in tables.

Analytics for "Configuração de Hardware" activity			
Analytics JSON: Download			
Student ID - 1002			
Qualitative Analytics - Check on provider			
#	Name	Type	Value
0	Acesso à atividade	boolean	true
1	Download Descritivo Técnico	boolean	true
2	Download Documentos Técnicos (%)	percentage	"50.00"
3	Progresso na atividade (%)	percentage	"66.67"
Total Completion			
Student ID - 1003			
Qualitative Analytics - Check on provider			
#	Name	Type	Value
0	Acesso à atividade	boolean	true
1	Download Descritivo Técnico	boolean	true
2	Download Documentos Técnicos (%)	percentage	"75.00"
3	Progresso na atividade (%)	percentage	"83.33"
Total Completion			

Figure 6.6: Activity Analytics view

6.2.4 Edit IAP view

This view is the core of the authoring tool frontend, enabling the user to manipulate an IAP. The component is shared for creating and editing, being capable of receiving a props object. If a previously saved IAP is loaded, the previously configured nodes are gathered from the backend and displayed, otherwise a blank canvas is displayed, as seen below on Figure 6.7.

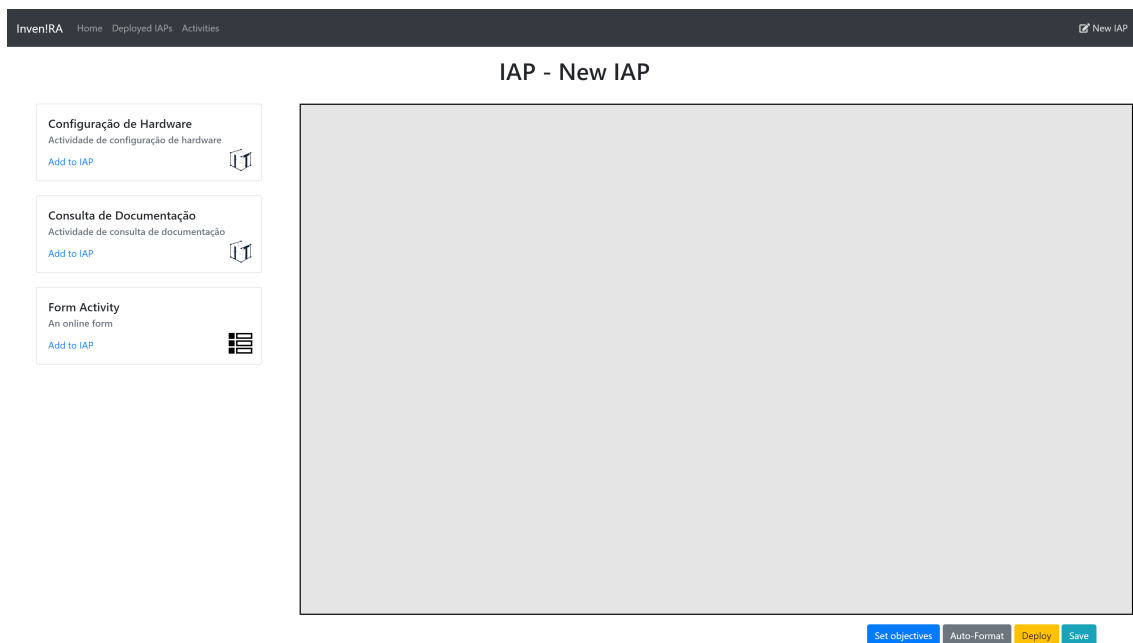


Figure 6.7: Adding an activity to the IAP modal

On this page, the available activities are displayed on the left, which are loaded from the backend and stored in the component state, followed by the prominent canvas and a few action buttons. The user is free to click “Add to IAP” on each activity or drag and drop the related icon to the canvas.

When an activity is added to the IAP, a configuration page is loaded on a modal, as seen on Figure 6.8. In this activity, we can see an embedded page developed by a third party for another project, which we will detail in Section 6.3.1.

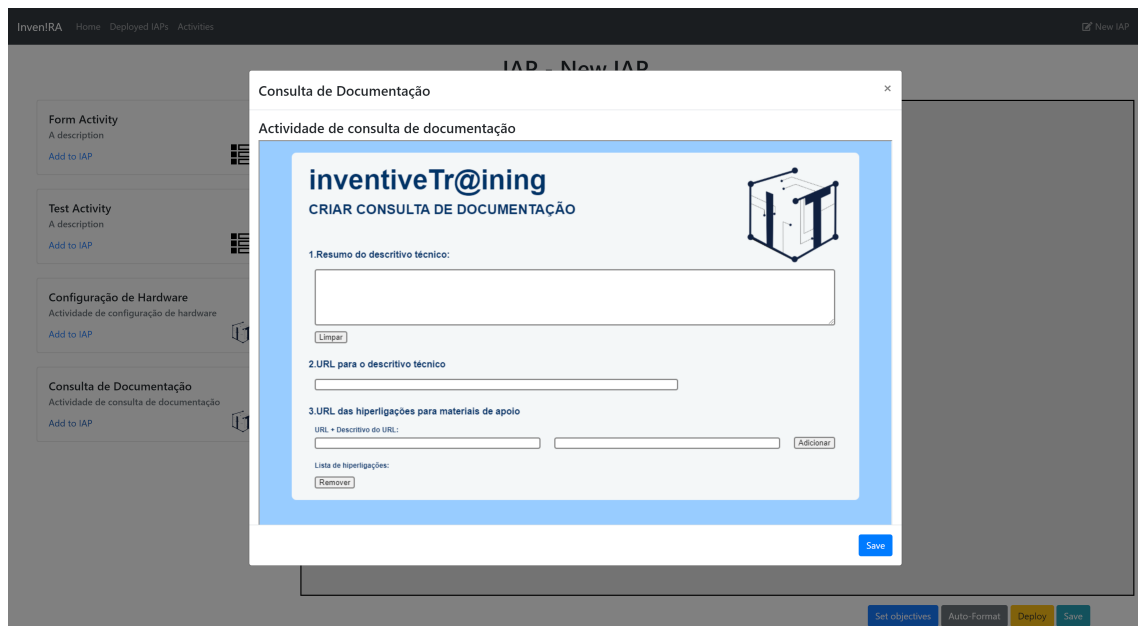


Figure 6.8: Adding an activity to the IAP modal

The information being displayed in the modal is the configuration page of the activity, as configured by the provider on activity creation. The user is then free to add configuration details and a JSON will be saved with this information, according to the configuration data.

When finished and saved, the activity is added to the graph, and all necessary data is saved in the object state. In Figure 6.9, we can see an IAP with some activities already inserted.

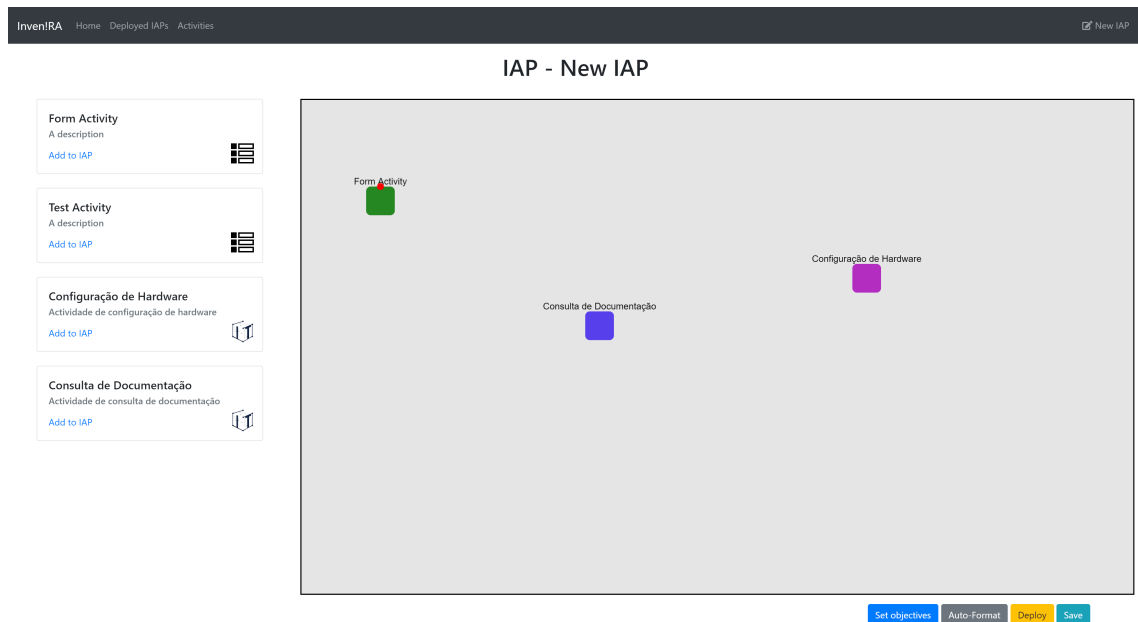


Figure 6.9: IAP with some activities

The authoring tool lets the user add edges to the graph, by dragging the red dot, which shows up on hovering or on single clicking a node, as seen on the green node on Figure ??, to another

node. The user can also auto-format the graph, for better visualization, as seen on Figure 6.10. This is purely for visualization purposes and the button cycles through some default sorting algorithms available on Cytoscape's API.

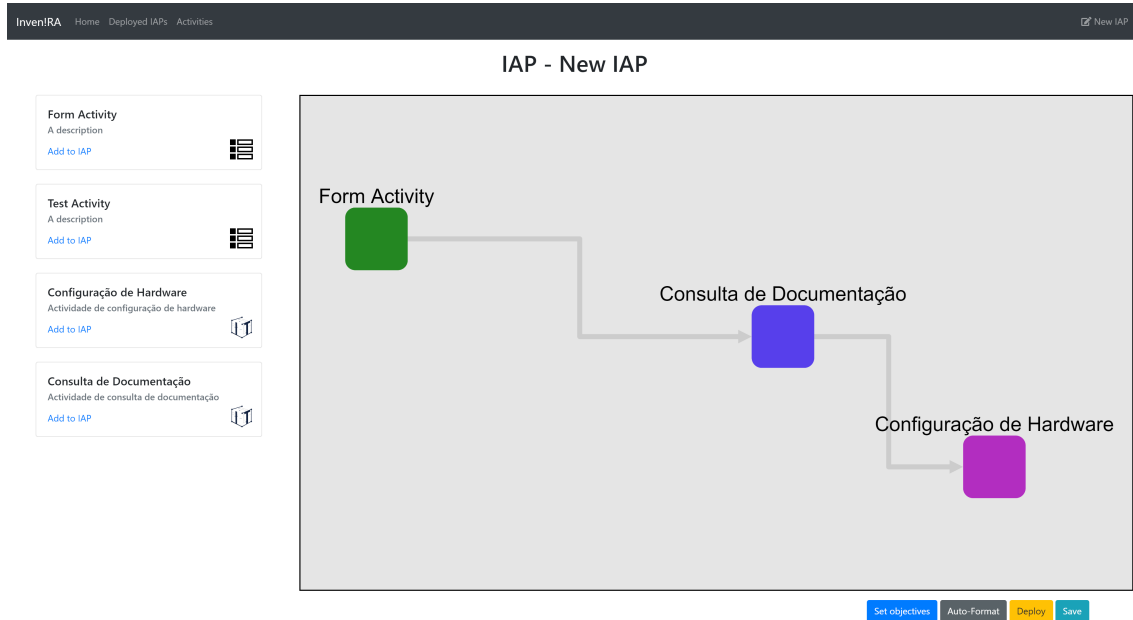


Figure 6.10: Auto-formatted IAP

Each node can be right-clicked, or long-pressed on touchscreens, to display a context menu with a few options, namely deletion, changing the colour of the node, or displaying the configuration page for that activity. This context menu is in the form of a circular menu surrounding the activity node, as can be seen on Figure 6.11.

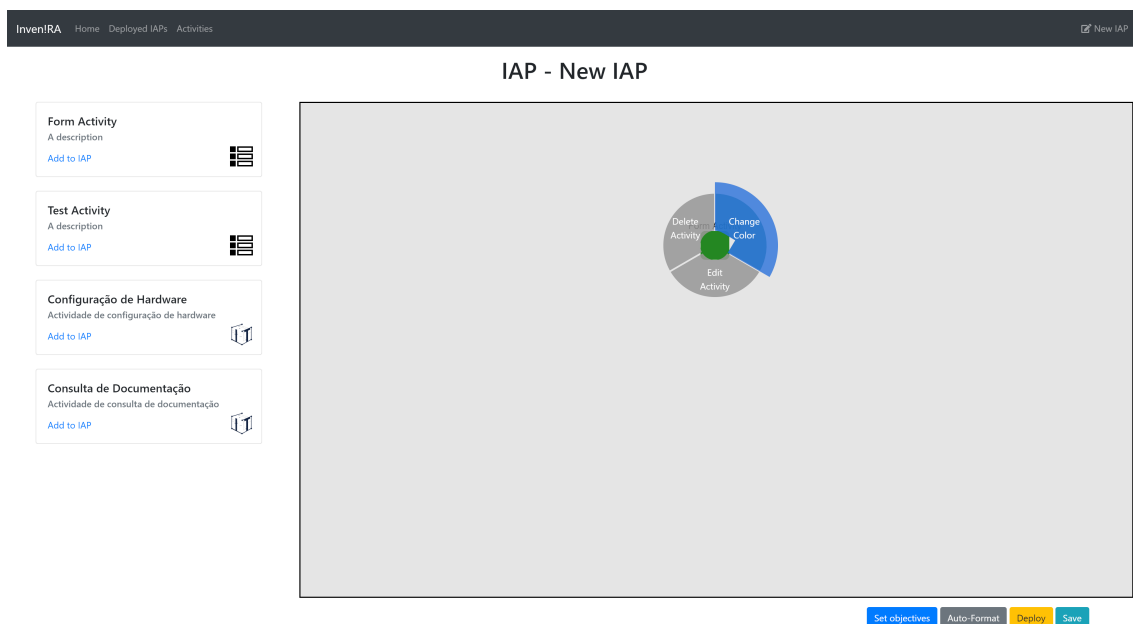


Figure 6.11: Activity Context Menu

By clicking on the “Set Objectives” button, the user can add learning objectives to the IAP. This action will pop-up a menu to set the objective details – this consists of a name and a set of analytics that contribute towards that objective, as seen on Figure 6.12.

The analytics shown are gathered from the activities already added to the IAP and can be selected at will from the available list. A weight for how much each analytic contributes to the objective can also be set in this menu, to be then reflected on how completion is computed.

The screenshot displays the 'IAP - New IAP' interface. On the left, there is a sidebar with three activity cards: 'Configuração de Hardware' (Actividade de configuração de hardware), 'Consulta de Documentação' (Actividade de consulta de documentação), and 'Form Activity' (An online form). Each card has an 'Add to IAP' button. The main area is titled 'IAP - New IAP' and contains a form titled 'Set the objectives for your IAP'. The form includes a text input for 'Your objective name' and a message: 'The following analytics were found in your activities. Please select the ones you want to track and their weight on the objective.' Below this, there are three sections of analytics with sliders for weight: 'Form Activity' (Complete form), 'Consulta de Documentação' (Access activity), and 'Download technical description document'. At the bottom of the form are two buttons: 'Set Objective' and 'Add Another Objective'. At the very bottom of the interface, there is a row of four buttons: 'Set objectives', 'Auto-Format', 'Deploy', and 'Save'.

Figure 6.12: Set objectives for IAP

6.2.5 IAP Analytics

From the deployed IAPs view, the user can load the analytics page for that IAP. In this page, the user can see the progress of the target users on each set objective, by gathering analytics from the relevant providers and compiling the data.

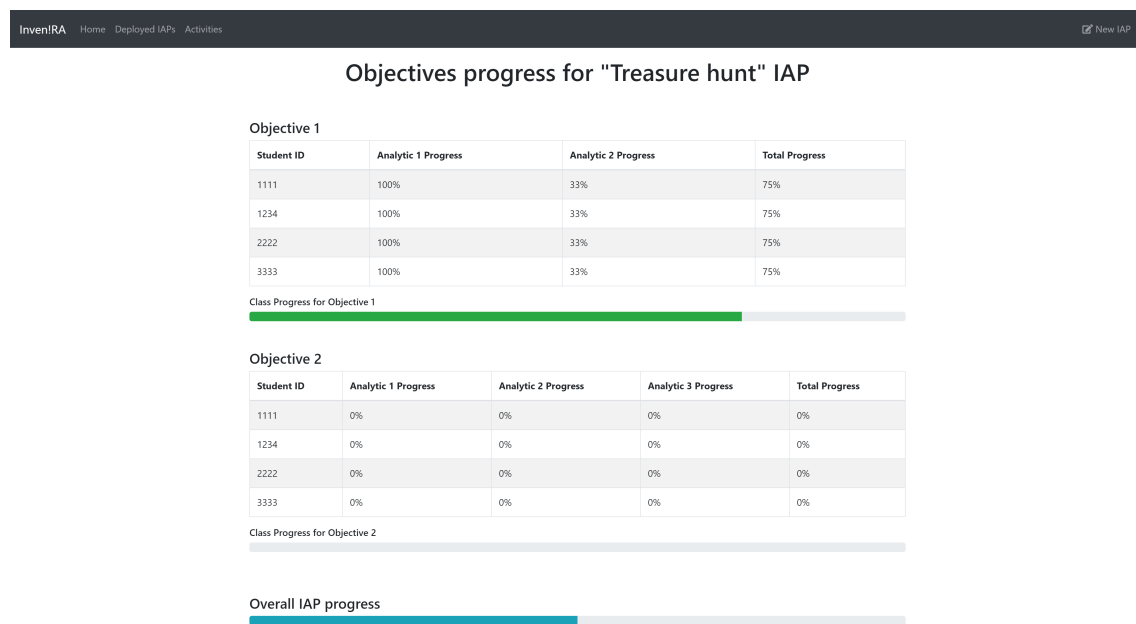


Figure 6.13: IAP objectives and analytics view

6.3 Integration Testing

In this section we will address the integration testing done, in the form of integrating with third party projects using this prototype for their own works. We will focus on Duarte Cota's inventiveTr@ining project, which had him act as a third-party activity provider for Inven!RA, and CHIC's implementation of an application for tourists, which makes use of Inven!RA's backend and concepts.

6.3.1 inventiveTr@ining by Duarte Cota

Duarte Cota is an Informatics Engineering student at the Aberta University. For his final project of the course, he developed the inventiveTr@ining project, a system for online tracking of micro-electronics projects. This work was supervised by Leonel Morgado, the same supervisor of the present work, which provided an opportunity to incorporate it under the Inven!RA umbrella.

The main objective of this work was to develop a tool for lecturers, which can help them develop and track activities in learning technological projects.

As it pertains to our project. Duarte played the part of a third-party activity provider, by designing activities that were registered in Inven!RA and available to configure and track in our prototype.

Figure 6.14: The configuration page of one of Duarte's developed activities

His work was paramount in the development of the JSON schemas and all necessary steps and requirements for successful integration with third parties, with important contributions on the deployment and analytics part of the work.

On Figure 6.15, we can see Duarte's sequence diagram for deploying an activity, which can better illustrate this process and how all parts connect.

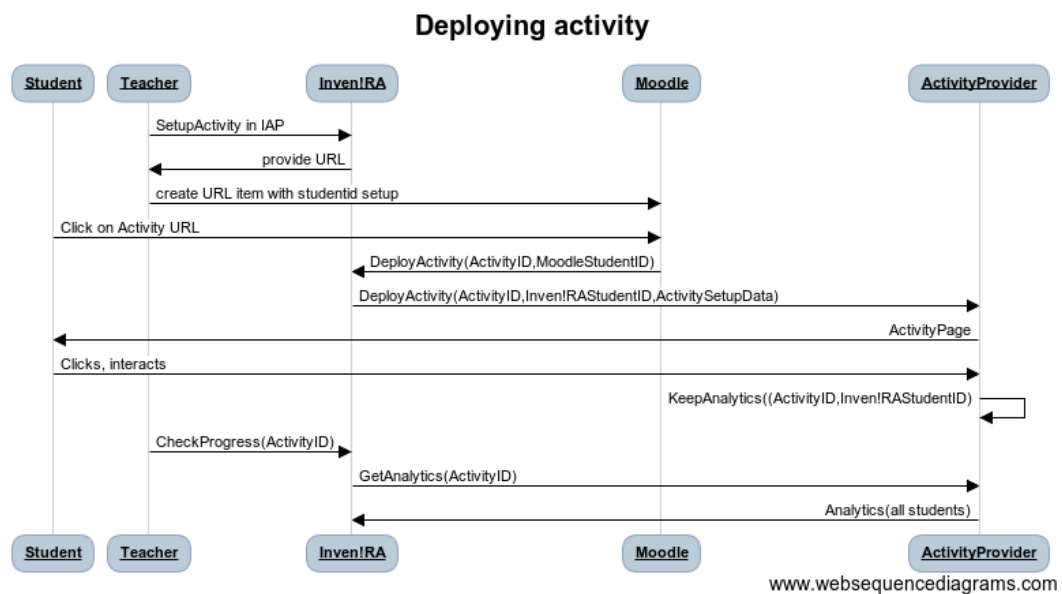


Figure 6.15: Activity Deployment Diagram from Duarte

On the scope of his work, Duarte created two activities in Inven!RA as proofs of concept. These can be seen and configured in the current prototype and helped us improve the experience of the prototype beyond sample activities.

Overall, integration was a success and the inclusion of Duarte’s work on Inven!RA was very important to the end result of the prototype. Duarte stressed, in his own report[17] that this integration was a major challenge, but a great learning opportunity is system integration. The main challenge came from the concurrent developments, which in a way helped both projects reach compromises, but also presented a challenge in understanding, defining and the compatibilization of calendars and ideas.

6.3.2 CHIC’s apps by Rui Quaresma

CHIC, which stands for Cooperative Holistic View on Internet and Content, is a project that aims to study and develop a digital system revolving around location-based activities, such as the exploration of touristic points of interest.

This project is a collaboration between several parties, such as the Faculty of Engineering of the University of Porto (FEUP), Gema Digital, the Catholic University of Portugal (UCP) and Jornal de Notícias (JN).

Beyond the institutions involved, the project per se is being developed mostly by students on their dissertation work, and Inven!RA takes an important part in the project, by providing its backend functionality, and lending its concepts.

The system developed by Rui Quaresma, our main point of contact, consists of two front-end applications, both developed as Progress Web Apps, using the Angular and Ionic JavaScript frameworks:

- An authoring tool - dedicated to authoring tourist experiences
- A mobile application – aimed at tourists, to enjoy these experiences

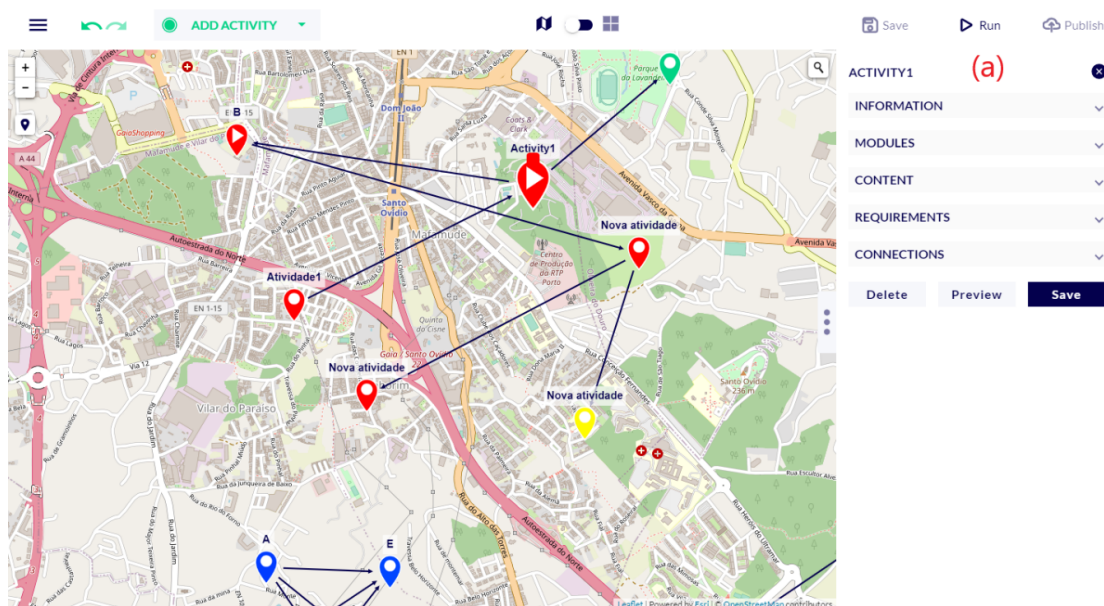


Figure 6.16: CHIC’s authoring tool

During the initial phase of both projects, ideas were still fuzzy, and cooperation was close between the teams. It soon became clear to us that CHIC's approach was quite different than ours, in that Inven!RA is a concept being studied for the long run, and CHIC is more of a commercial application. Since both projects required an authoring tool, we thought about sharing one with different functionalities, but CHIC's requirements and its inclusion of a design team and vastly different requirements made this impracticable. As our work focused on the backend, concept, and payloads, we decided that CHIC would leverage the backend, and all project-specific logic would be handled by the frontends. This proved to be the best approach, as the projects diverged.

It is then useful to revisit CHIC's development at the end of the proposed work and see how the integration worked. We can say it was mostly a success, with some caveats. As we were defining things for Inven!RA, CHIC's development was more design and user oriented, which made developments not always align.

The concepts of IAPs and activities were the main drivers of CHIC's implementation, as they were the first payloads we collaboratively set. The usage of a NoSQL database proved a major asset in integration, as the backend did not worry about any changes CHIC's team required, as long as the defined rules were adhered to.

Rui's feedback was very positive, in that the initial JSONs gave him a good basis for the work and not having to develop a backend and being abstracted of this part allowed him to focus on the most important parts of his work and make his development more agile. However, features that were more developed later on in Inven!RA, such as deployments, and features not fully implemented, such as user management, were largely ignored by CHIC. Notwithstanding, the solution to these limitations on Rui's part was always to handle things on his frontend, or use the non-linear structures in imaginative ways, and no other backend was needed, as Inven!RA provided all the necessary functionalities for both prototypes.

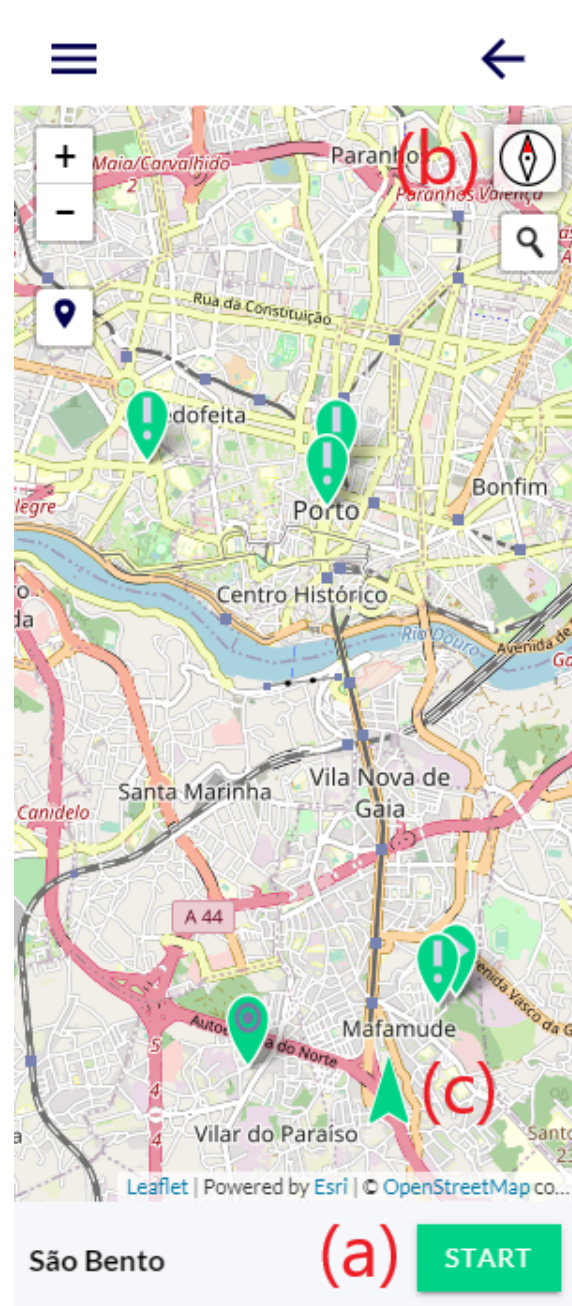


Figure 6.17: CHIC's mobile application for tourists

As a proof of concept for Inven!RA's backend, this project provided an incredible opportunity and constant feedback that helped improve our own work. Discussions with Rui and the rest of CHIC's team were very productive, and we ended up taking similar decisions and sharing knowledge. A keen eye will notice that, for example, the libraries used for node management are the same in our own frontend.

It is also relevant to mention that even execution for testing, as it pertained to the backend, was fully made available by the Inven!RA team, by running a separate instance of the backend on a separate port on our testing environment.

6.4 Results Summary

While the prototype and work presented in this document, such as the system architecture and expansion of the Inven!RA concept, are the most relevant results of this work, looking back at the testing done can provide us some insight into the quality of the developed artefacts.

The unit tests of the backend, as presented on Section 6.1, which were thoroughly ran throughout the development and before closing the code to write this document, give us confidence of the features implemented and their stability. The choice of frameworks such as *Moleculer* revealed very productive, with some error handling and server responses being automatically generated for cases we did not think on at the beginning, such as element deletion.

The visual testing done, along with validation of the service calls using Google Chrome's Developer Tools and the server logs, helped us better understand and improve the user interactions, and most importantly validate the backends' functions. It is relevant to mention that those tests were carried out in demos with the Inven!RA research team, and using the 3 major browsers in the market: Google Chrome V84, Mozilla Firefox V78 and Microsoft Edge V84 (Chromium Based). Successful testing was also performed on an Android Tablet, LG V8.3. Limited tests were done on Mobile devices, where all interactions worked but user experience and responsiveness of the frontend was lacking.

The integration with the *inventiveTr@ining* and CHIC projects were a major success of the project, and the most challenging part of the work, since it required strict collaboration with the partners, who at times needed specifications that we did not have. The end result, however, was extremely positive, with great feedback from the collaborators and successful integration into their own projects.

Chapter 7

Conclusions and future work

For this final chapter, we take a look back at the developed work, to reflect on the achieved objectives, the limitations of the prototype implementation, as well as address future work, leaving some suggestions.

7.1 Achieved objectives

Going back to Section 1.3, we set out to develop a solution that approached the importance of analytics in a learning environment, as well as providing an open ecosystem. We believe both those questions were handled in the prototype, as verified on Chapter 6. Analytics played a big part in how the system and payloads were designed and were a big focus when integrating with Duarte's project and in frontend visualization.

We believe that analytics can provide a lot of value in a solution such as Inven!RA, and we also think the research agrees. As for the open ecosystem question, the very existence of two successful integrations with independent projects proves its viability and relevance.

In sum, and focusing on Inven!RA as an ongoing and ambitious project, we believe there were significant contributes towards the following points:

- A solid specification for the system architecture
 - The architecture designed in this work, with special care on the backend functionality, provides a solid start for a more complex project and can easily be expanded. The specification of payloads and flows will likely be a strong foundation in any future work
- A working prototype for the authoring tool
 - Having a working prototype helps in understanding the real needs, in a way a concept just does not provide. We believe that our frontend prototype can be leveraged in future work, as it is a quite simple but capable application, that already tackled some of the

major hurdles that will come in the future of Inven!RA. The code can be expanded, but most of all the lessons learned will be carried over

- Successful integration with other projects
 - The integration with external projects proves the idea of an open system proposed by Inven!RA, where third party activity providers are essential

7.2 Limitations

While we believe the objectives set out on the start of the project were met, and that future work will handle a lot of our concerns, we would like to point out some points which we consider current limitations of the solution:

- Analytics combination
 - The proposed and implemented approach to analytics is quite simplistic, hindering the potential of this aspect of the solution. We believe analytics combination, and logic on top of the available analytics, such as predictive algorithms, should be explored. We believe that, as this topic is explored, limitations in the current approach will arise, and it will need to be revisited
- Open ecosystem was not subjected to stress of a diversity of different third-party situations and requirements
 - While testing with other real projects provided valuable insights, it is easy to fall into traps when you are focusing on a small set of data. When more providers come on board, the versatility and robustness of the proposed solution will truly be tested
- Large-scale testing may expose limitations of the concept
 - End-user testing was limited to the Inven!RA team and the teams of the two projects involved in this work. When large-scale testing is employed, and outside of the currently small ecosystem, limitations to the concept may be exposed

7.3 Future Work

Addressing future work for the Inven!RA platform is difficult, as it is still in its infancy and has so much potential. However, taking in consideration the authoring tool developed in this work and the objectives of the project, we can list some major things we identified and leave some suggestions.

- Develop Frontend that is representative of an end-user experience

- Our frontend is quite simple and just a proof of concept for the backend and the simplest use cases. For a rich end-user experience, a deep study on design and user experience should be performed, and all views should be carefully planned and optimized for optimal experience
- Complete and improve Learning Management System integration
 - Our host system module, which is responsible for LMS integration, is the most under-developed, as it was not a priority. Therefore, it only really provides a way to get some user information from Moodle, where ideally it would be rich in features and capable of plugging into different LMS platforms.
- Improve analytics tracking and visualization
 - The work on analytics, especially as it pertains to calculations of completion and its visualization, has a lot of headway for expansion. This topic is extremely vast, and we suspect will take a major role in Inven!RA moving forward.

Appendix A

Setup Guide

This guide takes the assumption you are interested in setting up all necessary components to run the Inven!RA Authoring Tool prototype. If you are interested in only in setting up one of the components, refer to its section.

It is also assumed that the reader has some familiarity with source code management, Node.js and basic terminal knowledge.

Please contact the team to obtain access to the source code repositories.

A.1 Backend

A.1.1 Pre-Requisites

1. MongoDB Instance
2. Node.js

You should start by setting up your MongoDB instance and gathering its connection string. The format should resemble the following:

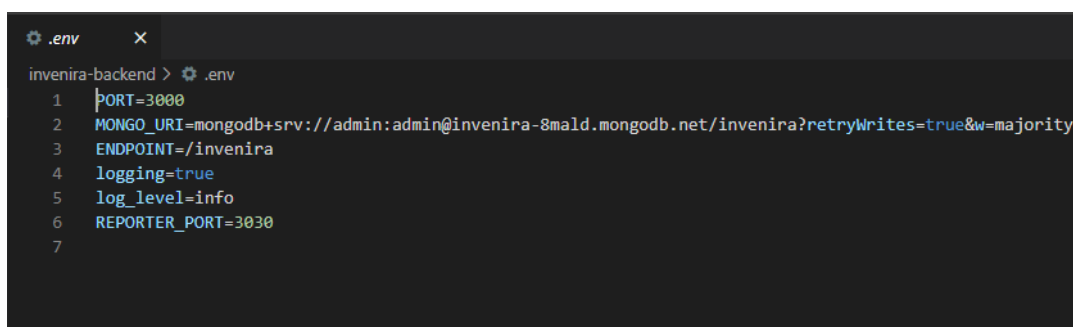
- `mongodb://[username:password@]host1[:port1][,...hostN[:portN]][/[defaultauth][?options]]`.

You should also clone the backend GIT repository to the intended machine and follow the steps below.

A.1.2 Installation steps

1. Open a terminal on the folder where you cloned the repository
2. Run "npm install" to install all necessary dependencies
3. Create a ".env" file in the root folder. You can customize the following environment variables:

- PORT - The port where the application will be exposed
- MONGO_URI - The connection string to the database
- ENDPOINT - The path where you want the application to be exposed
- LOGGING - Boolean to enable or disable verbose logging
- LOG_LEVEL - The level of detail for logging (info, warn, error)
- REPORTER_PORT - The port for Molecular's internal logger. Only change this if you're running multiple instances

A screenshot of a terminal window with a dark background. The title bar shows a gear icon, ".env", and a close button. The terminal content shows a prompt "invenira-backend >" followed by a sub-prompt ".env". Below this, seven lines of environment variables are listed, each preceded by a line number from 1 to 7. The variables are: PORT=3000, MONGO_URI=mongodb+srv://admin:admin@invenira-8mald.mongodb.net/invenira?retryWrites=true&w=majority, ENDPOINT=/invenira, logging=true, log_level=info, REPORTER_PORT=3030, and an empty line 7.

```
invenira-backend > .env
1 PORT=3000
2 MONGO_URI=mongodb+srv://admin:admin@invenira-8mald.mongodb.net/invenira?retryWrites=true&w=majority
3 ENDPOINT=/invenira
4 logging=true
5 log_level=info
6 REPORTER_PORT=3030
7
```

Figure A.1: Environment variables sample for the backend

The backend is now ready to run. To start it, run the "npm run molecular" command on the terminal.

A.2 Frontend

A.2.1 Pre-Requisites

1. Node.js
2. Service endpoint for a running backend

You should start by making sure you either are running the backend or were provided with an endpoint for a running backend.

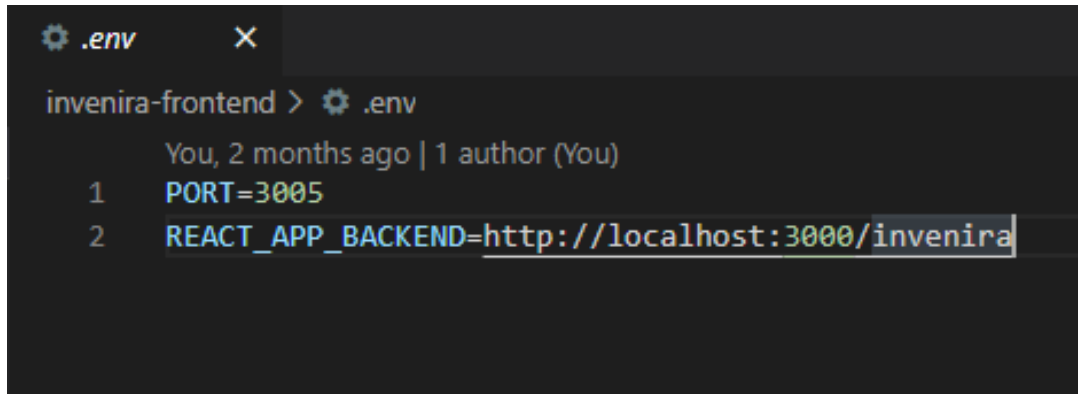
Clone the backend GIT repository to the intended machine and follow the steps below.

A.2.2 Installation steps

1. Open a terminal on the folder where you cloned the repository
2. Run "npm install" to install all necessary dependencies
3. The frontend source-code comes with an ".env" file by default. You can customize the following variables on it:

- (a) PORT - The port where the application will be exposed

(b) REACT_APP_BACKEND - The backend endpoint

A screenshot of a code editor window titled ".env" with a close button (X) in the top right corner. The editor shows the content of the ".env" file. At the top, it says "invenira-frontend > .env". Below that, there is a line of text: "You, 2 months ago | 1 author (You)". Then, there are two numbered lines: "1 PORT=3005" and "2 REACT_APP_BACKEND=http://localhost:3000/invenira". The second line is highlighted with a light blue background.

```
.env
invenira-frontend > .env
You, 2 months ago | 1 author (You)
1 PORT=3005
2 REACT_APP_BACKEND=http://localhost:3000/invenira
```

Figure A.2: Environment variables sample for the frontend

The frontend is now ready to run locally or be deployed to a webserver.

To run locally, run the "npm run start" command.

To create a production build, run "npm run build". The generated artefacts will be stored under the "build" folder, which you can expose on your webserver of choice.

A.3 Serving the application using NGINX

As it was the web server used, please find on Figure [A.3](#) the domain's "files-available" file to properly expose the application using NGINX.

This is only a sample approach and you can obviously choose whatever web server suits you and perform your own configurations.

The approach is compatible with multiple backends and frontends. You should change "example.com" with your domain.

```

1 server {
2     listen 80;
3     listen [::]:80;
4
5     server_name example.com www.example.com;
6
7     gzip on;
8     gzip_disable "msie6";
9     gzip_vary on;
10    gzip_proxied any;
11    gzip_comp_level 6;
12    gzip_buffers 16 8k;
13    gzip_http_version 1.1;
14    gzip_min_length 0;
15    gzip_types text/plain text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript application/vnd.ms-fontobject application/x-font-ttf font/opentype;
16
17    location /invenira/ {
18        proxy_pass http://localhost:3000/;
19    }
20
21    location /backend2/ {
22        proxy_pass http://localhost:3001/;
23    }
24
25    location ~ /invenira-demo/ {
26        alias /var/www/example.com/invenira-frontend/build/;
27        index index.html index.htm;
28        #rewrite "/invenira-demo/(.*)" /invenira-demo/$1;
29        try_files $uri $uri/ /invenira-demo/index.html$args;
30    }
31
32    location /files/ {
33        root /var/www/example.com;
34        autoindex on;
35        if ($request_method = 'GET') {
36            add_header 'Access-Control-Allow-Origin' '*';
37            add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';
38            add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range';
39            add_header 'Access-Control-Expose-Headers' 'Content-Length,Content-Range';
40        }
41    }
42
43    location / {
44        root /var/www/example.com/html;
45        index index.html index.htm;
46    }
47
48    location ~ /\.ht {
49        deny all;
50    }
51
52 }

```

Figure A.3: NGINX sample configuration

A few notes on the approach done:

1. The backend(s) is running locally on the server and being exposed through proxy-pass
2. The frontend needs to be built using "npm run build"
3. CORS must be enabled

Appendix B

REST Specification

This document specifies the currently exposed REST services for the Inven!RA backend as well as sample payloads for each operation's request and response. On Section [B.3](#) you can find the full JSON Schemas for the 2 most relevant objects, IAP and Activity.

To improve readability, the samples were linked in each operation and aggregated in section [B.2](#).

B.1 REST API

Inven!RA backend path: `%host%:%port%/invenira/`

1. Local default: `http://localhost:3000/invenira/`
2. Test environment `http://tiagoc.xyz:80/invenira/`

B.1.1 IAP Services

The services below enable IAP manipulation.

B.1.1.1 List IAPs

This service returns a list of all available IAPs.

Relative path: `/iap`

Method: GET

Request Parameter: N.A.

Request Payload: N.A.

Response: Array of IAPs

Sample Request: N.A.

Sample Response: [B.2.1](#)

B.1.1.2 Get IAP by ID

This service returns an IAP by its ID.

Relative path: [/iap/{id}](#)

Method: GET

Request Parameter: id

Request Payload: N.A.

Response: IAP object or 404

Sample Request: N.A.

Sample Response: [B.2.2](#)

B.1.1.3 Create IAP

Creates an IAP object.

Relative path: [/iap](#)

Method: POST

Request Parameter: N.A.

Request Payload: IAP Object

Response: IAP object with id

Sample Request: [B.2.3](#)

Sample Response: [B.2.2](#)

B.1.1.4 Update IAP

Updates an IAP object.

Relative path: [/iap/{id}](#)

Method: PUT

Request Parameter: id

Request Payload: IAP Object

Response: IAP object with id

Sample Request: [B.2.3](#)

Sample Response: [B.2.2](#)

B.1.1.5 Delete IAP

Deletes an IAP object.

Relative path: [/iap/{id}](#)

Method: DELETE

Request Parameter: id

Request Payload: N.A.

Response: Success or 404

Sample Request: [B.2.3](#)

Sample Response: [B.2.2](#)

B.1.2 Activity Services

The services below enable activity manipulation.

B.1.2.1 List activities

This service returns a list of all available activities.

Relative path: [/activity](#)

Method: GET

Request Parameter: N.A.

Request Payload: N.A.

Response: Array of activities

Sample Request: N.A.

Sample Response: [B.2.4](#)

B.1.2.2 Get activity by ID

This service returns an activity by its ID.

Relative path: [/activity/{id}](#)

Method: GET

Request Parameter: id

Request Payload: N.A.

Response: Activity object or 404

Sample Request: N.A.

Sample Response: [B.2.5](#)

B.1.2.3 Create activity

Creates an activity object.

Relative path: [/activity](#)

Method: POST

Request Parameter: N.A.

Request Payload: Activity object

Response: Activity object with id

Sample Request: [B.2.6](#)

Sample Response: [B.2.5](#)

B.1.2.4 Update activity

Updates an activity object.

Relative path: [/activity/{id}](#)

Method: PUT

Request Parameter: id

Request PayloadActivity object

ResponseActivity object with id

Sample Request: [B.2.6](#)

Sample Response: [B.2.5](#)

B.1.2.5 Delete activity

Deletes an activity object.

Relative path: [/activity/{id}](#)

Method: DELETE

Request Parameter: id

Request Payload: N.A.

Response: Success or 404

Sample Request: [B.2.6](#)

Sample Response: [B.2.5](#)

B.1.3 Deployment Services

The services below relate to IAP Deployment.

B.1.3.1 List Users

This service returns a list of users.

Relative path: [/users](#)

Method: GET

Request Parameter: N.A.

Request Payload: N.A.

Response: Array of Users

Sample Request: N.A.

Sample Response: [B.2.9](#)

B.1.3.2 List Deployed IAPs

This service returns a list of all deployed IAPs.

Relative path: [/deploy/iap](#)

Method: GET

Request Parameter: N.A.

Request Payload: N.A.

Response: Array of IAPs

Sample Request: N.A.

Sample Response: [B.2.1](#)

B.1.3.3 Deploy IAP

Creates a deployed IAP object.

Relative path: [/deploy](#)

Method: POST

Request Parameter: N.A.

Request Payload: Valid IAP Object with objectives array

Response: IAP object with id

Sample Request: [B.2.7](#)

Sample Response: [B.2.2](#)

B.1.3.4 Get Deployed IAP by ID

This service returns a deployed IAP by its ID.

Relative path: [/deploy/iap/{id}](#)

Method: GET

Request Parameter: id

Request Payload: N.A.

Response: IAP object or 404

Sample Request: N.A.

Sample Response: [B.2.2](#)

B.1.3.5 List Deployed Activities for IAP

This service returns a list of all activities in a deployed IAP.

Relative path: [/deploy/activities/{id}](#)

Method: GET

Request Parameter: id - deployed iap id

Request Payload: N.A.

Response: Array of activities with deployment urls

Sample Request: N.A.

Sample Response: [B.2.8](#)

B.1.3.6 Trigger deployed activity

This service handles the start of an activity and redirects to experience url

Relative path: [/deploy/activity/?id={id}&userId={userId}](#)

Method: GET

Request Parameter: id - activity id, userId - lms user id

Request Payload: N.A.

Response: Redirect to provider url

Sample Request: N.A.

Sample Response: N.A.

B.2 Sample Payloads

B.2.1 Sample Response List IAPs

```
1  [
2    {
3      "_id": "5eb0b6410e0ac238c4e8e624",
4      "name": "My Test IAP",
5      "properties": [
6        {
7          "description": "A sample description"
8        }
9      ],
10     "nodes": [
11       {
12         "node_id": "act-1",
13         "act_id": "5eb0b6100e0ac238c4e8e623",
14         "name": "Test Activity",
15         "properties": [
16           {
17             "pos_x": 150,
18             "pos_y": 150
19           }
20         ]
21       },
22       {
23         "node_id": "act-2",
24         "act_id": "5eb0b5110e0ac238c4e8e621",
25         "name": "DuarteLda Actividade Test Html Internal",
26         "properties": [
27           {
28             "pos_x": 400,
29             "pos_y": 150
30           }
31         ]
32       },
```

```
33     {
34         "node_id": "act-3",
35         "act_id": "5eb0b5f50e0ac238c4e8e622",
36         "name": "Form Activity",
37         "properties": [
38             {
39                 "pos_x": 400,
40                 "pos_y": 450
41             }
42         ]
43     },
44     {
45         "node_id": "64af367e-8602-463f-9aab-645dd2e60362",
46         "act_id": "5eb0b6f70e0ac238c4e8e625",
47         "name": "DuarteLda Activity",
48         "properties": [
49             {
50                 "pos_x": 747,
51                 "pos_y": 393
52             }
53         ]
54     },
55     {
56         "node_id": "95de2ce1-feff-4803-b830-b1c4ac2cb9a9",
57         "act_id": "5eb0b6f70e0ac238c4e8e625",
58         "name": "DuarteLda Activity",
59         "properties": [
60             {
61                 "pos_x": 1021,
62                 "pos_y": 764
63             }
64         ]
65     }
66 ],
67 "edges": [
68     {
69         "source": "act-1",
70         "target": "act-2",
71         "label": "",
72         "id": "e81056a0-fd42-4990-971b-8226a384ee31"
```

```

73     }
74   ]
75 },
76 {
77   "_id": "5eb0d3870e0ac238c4e8e626",
78   "name": "New IAP",
79   "properties": [
80     {
81       "description": "This is a new IAP"
82     }
83   ],
84   "nodes": [
85     {
86       "node_id": "daac7cf3-c873-4e7f-861a-2cea134de3a9",
87       "act_id": "5eb0b6100e0ac238c4e8e623",
88       "name": "Test Activity",
89       "properties": [
90         {
91           "pos_x": 109,
92           "pos_y": 102
93         }
94       ]
95     }
96   ],
97   "edges": []
98 ]
99 }
100 ]

```

B.2.2 Sample Response Get IAP

```

1  {
2    "_id": "5eee571ea691f8225432699a",
3    "name": "Porto Exploration",
4    "properties": [
5      {
6        "description": "This is an IAP with LBS activities"
7      }
8    ],

```



```
9     "nodes": [  
10         {  
11             "node_id": "act-1",  
12             "act_id": "5eb0b6100e0ac238c4e8e623",  
13             "name": "One",  
14             "properties": [  
15                 {  
16                     "pos_x": 150,  
17                     "pos_y": 150  
18                 }  
19             ]  
20         },  
21         {  
22             "node_id": "act-2",  
23             "act_id": "5eb0b5110e0ac238c4e8e621",  
24             "name": "Two",  
25             "properties": [  
26                 {  
27                     "pos_x": 400,  
28                     "pos_y": 150  
29                 }  
30             ]  
31         },  
32         {  
33             "node_id": "act-3",  
34             "act_id": "5eb0b5f50e0ac238c4e8e622",  
35             "name": "Three",  
36             "properties": [  
37                 {  
38                     "pos_x": 400,  
39                     "pos_y": 450  
40                 }  
41             ]  
42         }  
43     ],  
44     "edges": [  
45         {  
46             "source": "act-1",  
47             "target": "act-2",  
48             "label": ""
```

```
49     }
50   ]
51 }
```

B.2.3 Sample Request Create IAP

```
1  {
2      "name": "My Test IAP",
3      "properties": [
4          {
5              "description": "A sample description"
6          }
7      ],
8      "nodes": [
9          {
10             "node_id": "act-1",
11             "act_id": "5eb0b6100e0ac238c4e8e623",
12             "name": "One",
13             "properties": [
14                 {
15                     "pos_x": 150,
16                     "pos_y": 150
17                 }
18             ]
19         },
20         {
21             "node_id": "act-2",
22             "act_id": "5eb0b5110e0ac238c4e8e621",
23             "name": "Two",
24             "properties": [
25                 {
26                     "pos_x": 400,
27                     "pos_y": 150
28                 }
29             ]
30         },
31         {
32             "node_id": "act-3",
33             "act_id": "5eb0b5f50e0ac238c4e8e622",
```

```

34         "name": "Three",
35         "properties": [
36             {
37                 "pos_x": 400,
38                 "pos_y": 450
39             }
40         ]
41     },
42 ],
43     "edges": [
44         {
45             "source": "act-1",
46             "target": "act-2",
47             "label": ""
48         }
49     ]
50 }

```

B.2.4 Sample Request List Activities

```

1  [
2    {
3      "_id": "5eb0b5110e0ac238c4e8e621",
4      "name": "Form Activity",
5      "properties": [
6        {
7          "icon": "http://tiagoc.xyz/files/form.png",
8          "description": "A description"
9        }
10     ],
11     "config_url": "http://tiagoc.xyz/files/form.html",
12     "json_params": "http://tiagoc.xyz/files/config_fields.json",
13     "user_url": "http://tiagoc.xyz/files/student.html",
14     "analytics": "http://tiagoc.xyz/files/analytics_2.json"
15   },
16   {
17     "_id": "5eb0b5f50e0ac238c4e8e622",
18     "name": "DuarteLda Actividade Test Html Internal",
19     "properties": [

```

```

20     {
21         "icon":
22             ↪ "http://tiagoc.xyz/files/duartecota/img/it_logo.png",
23         "description": "This is Duarte's test activity"
24     },
25     "config_url":
26         ↪ "http://tiagoc.xyz/files/duartecota/activity_create_doc.html",
27     "json_params":
28         ↪ "http://tiagoc.xyz/files/duartecota/config_fields.json",
29     "user_url":
30         ↪ "http://tiagoc.xyz/files/duartecota/activity_create_doc.html",
31     "analytics": []
32 },
33 {
34     "_id": "5eb0b6100e0ac238c4e8e623",
35     "name": "Test Activity",
36     "properties": [
37         {
38             "icon": "http://tiagoc.xyz/files/form.png",
39             "description": "A description"
40         }
41     ],
42     "config_url": "http://tiagoc.xyz/files/form.html",
43     "json_params": "http://tiagoc.xyz/files/config_fields.json",
44     "user_url": "http://tiagoc.xyz/files/student.html",
45     "analytics": "http://tiagoc.xyz/files/analytics.json"
46 },
47 {
48     "_id": "5ee285795ea9211d18132574",
49     "name": "Duarte Lda. Activity",
50     "properties": [
51         {
52             "icon": "http://www.duartecota.com/img/it_logo.png",
53             "description": "This is Duarte's test activity on his
54                 ↪ server"
55         }
56     ],
57     "config_url":
58         ↪ "http://www.duartecota.com/html/activity_config_doc.html",

```

```
54     "json_params":
55       ↪ "http://tiagoc.xyz/files/duartecota/config_fields.json",
56     "user_url":
57       ↪ "http://www.duartecota.com/html/activity_config_hard.html",
58     "analytics":
59       ↪ "http://tiagoc.xyz/files/duartecota/exemplo_atividade1_deploy.json"
60   },
61   {
62     "_id": "5ee426185ea9211d18132578",
63     "name": "Configuração de Hardware",
64     "properties": [
65       {
66         "icon": "http://www.duartecota.com/img/it_logo.png",
67         "description": "Atividade de configuração de hardware"
68       }
69     ],
70     "config_url":
71       ↪ "https://inventivetraining.herokuapp.com/configHardActivity",
72     "json_params":
73       ↪ "http://tiagoc.xyz/files/duartecota/hardware_duarte_deploy_sample.json",
74     "user_url":
75       ↪ "https://inventivetraining.herokuapp.com/api/deployHardRequestRoute",
76     "analytics":
77       ↪ "http://tiagoc.xyz/files/duarte_cota/analytics_duarte.json"
78   },
79   {
80     "_id": "5ee426505ea9211d18132579",
81     "name": "Consulta de Documentação",
82     "properties": [
83       {
84         "icon": "http://www.duartecota.com/img/it_logo.png",
85         "description": "Atividade de consulta de documentação"
86       }
87     ],
88     "config_url":
89       ↪ "https://inventivetraining.herokuapp.com/configDocActivity",
90     "json_params":
91       ↪ "http://tiagoc.xyz/files/duartecota/doc_duarte_deploy_sample.json",
92     "user_url":
93       ↪ "https://inventivetraining.herokuapp.com/api/deployDocRequestRoute",
```

```
84     "analytics":  
      ↪ "http://tiagoc.xyz/files/duarte_cota/analytics_duarte.json"  
85   }  
86 ]
```

B.2.5 Sample Request Create Activity

```
1  {  
2      "name": "Consulta de Documentação",  
3      "properties": [  
4          {  
5              "icon":  
              ↪ "http://www.duarte_cota.com/img/it_logo.png",  
6              "description": "Actividade de consulta de  
              ↪ documentação"  
7          }  
8      ],  
9      "config_url":  
      ↪ "https://inventivetraining.herokuapp.com/configDocActivity",  
10     "json_params":  
      ↪ "http://tiagoc.xyz/files/duartecota/doc_duarte_deploy_sample.json",  
11     "user_url":  
      ↪ "https://inventivetraining.herokuapp.com/api/deployDocRequestRoute",  
12     "analytics":  
      ↪ "http://tiagoc.xyz/files/duarte_cota/analytics_duarte.json"  
13 }
```

B.2.6 Sample Request Get Activity

```
1  {  
2      "_id": "5ef8157c82d078e2703fdc82",  
3      "name": "Form Activity",  
4      "properties": [  
5          {  
6              "icon": "http://tiagoc.xyz/files/form.png",  
7              "description": "An online form"  
8          }  
9      ],  
10     "config_url": "http://tiagoc.xyz/files/form.html",
```

```
11  "json_params": "http://tiagoc.xyz/files/config_fields.json",
12  "user_url": "http://tiagoc.xyz/files/student.html",
13  "analytics": "http://tiagoc.xyz/files/analytics_2.json"
14 }
```

B.2.7 Sample deploy IAP

```
1  {
2      "name": "Treasure hunt",
3      "properties": [
4          {
5              "description": "Alice in Wonderland
6                  ↪ Adventure"
7          }
8      ],
9      "nodes": [
10         {
11             "node_id": "act-1",
12             "act_id": "5eb0b6100e0ac238c4e8e623",
13             "config_json": "url.json",
14             "name": "One",
15             "properties": [
16                 {
17                     "pos_x": 150,
18                     "pos_y": 150
19                 }
20             ],
21         },
22         {
23             "node_id": "act-2",
24             "act_id": "5eb0b5110e0ac238c4e8e621",
25             "name": "Two",
26             "config_json": "url.json",
27             "properties": [
28                 {
29                     "pos_x": 400,
30                     "pos_y": 150
31                 }
32             ]
33         }
34     ]
35 }
```

```

32         },
33         {
34             "node_id": "act-3",
35             "act_id": "5eb0b5f50e0ac238c4e8e622",
36             "name": "Three",
37             "config_json": "url.json",
38             "properties": [
39                 {
40                     "pos_x": 400,
41                     "pos_y": 450
42                 }
43             ]
44         }
45     ],
46     "edges": [
47         {
48             "source": "act-1",
49             "target": "act-2",
50             "label": ""
51         }
52     ],
53     "objectives": [
54         {
55             "name": "Use form",
56             "analytics": [
57                 {
58                     "name": "Complete form",
59                     "id": "101"
60                 }
61             ]
62         },
63         {
64             "name": "Complete documentation class",
65             "analytics": [
66                 {
67                     "name": "Access activity",
68                     "id": "102"
69                 },
70                 {

```



```
71         "name": "Download technical
72         ↪ description document",
73         "id": "103"
74     }
75 ]
76 }
77 }
```

B.2.8 Sample list activities for deployed IAP

```
1  [
2    {
3      "_id": "123",
4      "name": "Consulta de Documentação",
5      "deployment_url":
6      ↪ "http://tiagoc.xyz/invenira/deploy/activity?id=123"
7    },
8    {
9      "_id": "1234",
10     "name": "Configuração de Hardware",
11     "deployment_url":
12     ↪ "http://tiagoc.xyz/invenira/deploy/activity?id=1234"
13   }
14 ]
```

B.2.9 Sample list activities for deployed IAP

```
1  [
2    {
3      "user_id": "1",
4      "moodle_id": "5",
5      "groups": []
6    },
7    {
8      "user_id": "2",
9      "moodle_id": "4",
10     "groups": []
11   }
12 ]
```

```
11     }
12 ]
```

B.3 JSON Schemas

B.3.1 IAP Schema

```
1  {
2    "$schema": "http://json-schema.org/draft-07/schema",
3    "$id": "http://example.com/example.json",
4    "type": "object",
5    "title": "The root schema",
6    "description": "The root schema comprises the entire JSON
7    ↪ document.",
8    "default": {},
9    "examples": [
10      {
11        "_id": "5eee571ea691f8225432699a",
12        "name": "Porto Exploration",
13        "properties": [
14          {
15            "description": "This is an IAP with LBS
16            ↪ activities"
17          }
18        ],
19        "nodes": [
20          {
21            "node_id": "act-1",
22            "act_id": "5eb0b6100e0ac238c4e8e623",
23            "name": "One",
24            "properties": [
25              {
26                "pos_x": 150,
27                "pos_y": 150
28              }
29            ]
30          },
31          {
32            "node_id": "act-2",
```

```

31         "act_id": "5eb0b5110e0ac238c4e8e621",
32         "name": "Two",
33         "properties": [
34             {
35                 "pos_x": 400,
36                 "pos_y": 150
37             }
38         ]
39     },
40     {
41         "node_id": "act-3",
42         "act_id": "5eb0b5f50e0ac238c4e8e622",
43         "name": "Three",
44         "properties": [
45             {
46                 "pos_x": 400,
47                 "pos_y": 450
48             }
49         ]
50     }
51 ],
52 "edges": [
53     {
54         "source": "act-1",
55         "target": "act-2",
56         "label": ""
57     }
58 ]
59 },
60 ],
61 "required": [
62     "_id",
63     "name",
64     "properties",
65     "nodes",
66     "edges"
67 ],
68 "additionalProperties": true,
69 "properties": {
70     "_id": {

```

```

71     "$id": "#/properties/_id",
72     "type": "string",
73     "title": "The _id schema",
74     "description": "An explanation about the purpose of this
    ↪ instance.",
75     "default": "",
76     "examples": [
77         "5eee571ea691f8225432699a"
78     ]
79 },
80 "name": {
81     "$id": "#/properties/name",
82     "type": "string",
83     "title": "The name schema",
84     "description": "An explanation about the purpose of this
    ↪ instance.",
85     "default": "",
86     "examples": [
87         "Porto Exploration"
88     ]
89 },
90 "properties": {
91     "$id": "#/properties/properties",
92     "type": "array",
93     "title": "The properties schema",
94     "description": "An explanation about the purpose of this
    ↪ instance.",
95     "default": [],
96     "examples": [
97         [
98             {
99                 "description": "This is an IAP with LBS
    ↪ activities"
100             }
101         ]
102     ],
103     "additionalItems": true,
104     "items": {
105         "anyOf": [
106             {

```

```

107         "$id":
108             ↪ "#/properties/properties/items/anyOf/0",
109         "type": "object",
110         "title": "The first anyOf schema",
111         "description": "An explanation about the
112             ↪ purpose of this instance.",
113         "default": {},
114         "examples": [
115             {
116                 "description": "This is an IAP with
117                     ↪ LBS activities"
118             }
119         ],
120         "required": [
121             "description"
122         ],
123         "additionalProperties": true,
124         "properties": {
125             "description": {
126                 "$id":
127                     ↪ "#/properties/properties/items/anyOf/0/prop
128                 "type": "string",
129                 "title": "The description schema",
130                 "description": "An explanation about
131                     ↪ the purpose of this instance.",
132                 "default": "",
133                 "examples": [
134                     "This is an IAP with LBS
135                     ↪ activities"
136                 ]
137             }
138         }
139     ],
140     "$id": "#/properties/properties/items"
141 }
142 },
143 "nodes": {
144     "$id": "#/properties/nodes",
145     "type": "array",

```

```

141     "title": "The nodes schema",
142     "description": "An explanation about the purpose of this
    ↪ instance.",
143     "default": [],
144     "examples": [
145         [
146             {
147                 "node_id": "act-1",
148                 "act_id": "5eb0b6100e0ac238c4e8e623",
149                 "name": "One",
150                 "properties": [
151                     {
152                         "pos_x": 150,
153                         "pos_y": 150
154                     }
155                 ]
156             },
157             {
158                 "node_id": "act-2",
159                 "act_id": "5eb0b5110e0ac238c4e8e621",
160                 "name": "Two",
161                 "properties": [
162                     {
163                         "pos_x": 400,
164                         "pos_y": 150
165                     }
166                 ]
167             }
168         ]
169     ],
170     "additionalItems": true,
171     "items": {
172         "anyOf": [
173             {
174                 "$id": "#/properties/nodes/items/anyOf/0",
175                 "type": "object",
176                 "title": "The first anyOf schema",
177                 "description": "An explanation about the
    ↪ purpose of this instance.",
178                 "default": {},

```

```

179     "examples": [
180         {
181             "node_id": "act-1",
182             "act_id":
183                 ↪ "5eb0b6100e0ac238c4e8e623",
184             "name": "One",
185             "properties": [
186                 {
187                     "pos_x": 150,
188                     "pos_y": 150
189                 }
190             ]
191         },
192     "required": [
193         "node_id",
194         "act_id",
195         "name",
196         "properties"
197     ],
198     "additionalProperties": true,
199     "properties": {
200         "node_id": {
201             "$id":
202                 ↪ "#/properties/nodes/items/anyOf/0/properties
203             "type": "string",
204             "title": "The node_id schema",
205             "description": "An explanation about
206                 ↪ the purpose of this instance.",
207             "default": "",
208             "examples": [
209                 "act-1"
210             ]
211         },
212         "act_id": {
213             "$id":
214                 ↪ "#/properties/nodes/items/anyOf/0/properties
215             "type": "string",
216             "title": "The act_id schema",

```

```

214     "description": "An explanation about
      ↪ the purpose of this instance.",
215     "default": "",
216     "examples": [
217         "5eb0b6100e0ac238c4e8e623"
218     ]
219 },
220 "name": {
221     "$id":
222     ↪ "#/properties/nodes/items/anyOf/0/properties/name",
223     "type": "string",
224     "title": "The name schema",
225     "description": "An explanation about
      ↪ the purpose of this instance.",
226     "default": "",
227     "examples": [
228         "One"
229     ]
230 },
231 "properties": {
232     "$id":
233     ↪ "#/properties/nodes/items/anyOf/0/properties/properties",
234     "type": "array",
235     "title": "The properties schema",
236     "description": "An explanation about
      ↪ the purpose of this instance.",
237     "default": [],
238     "examples": [
239         [
240             {
241                 "pos_x": 150,
242                 "pos_y": 150
243             }
244         ]
245     ],
246     "additionalItems": true,
247     "items": {
248         "anyOf": [
249             {

```



```

248 "$id":
    ↪ "#/properties/nodes/items/anyOf/
249 "type": "object",
250 "title": "The first
    ↪ anyOf schema",
251 "description": "An
    ↪ explanation about
    ↪ the purpose of this
    ↪ instance.",
252 "default": {},
253 "examples": [
254     {
255         "pos_x": 150,
256         "pos_y": 150
257     }
258 ],
259 "required": [
260     "pos_x",
261     "pos_y"
262 ],
263 "additionalProperties":
    ↪ true,
264 "properties": {
265     "pos_x": {
266         "$id":
            ↪ "#/properties/nodes/item
267         "type":
            ↪ "integer",
268         "title": "The
            ↪ pos_x
            ↪ schema",
269         "description":
            ↪ "An
            ↪ explanation
            ↪ about the
            ↪ purpose of
            ↪ this
            ↪ instance.",
270         "default": 0,
271         "examples": [

```

```

272                                     150
273                                 ]
274                             },
275                             "pos_y": {
276                                 "$id":
277                                     ↪ "#/properties/nodes/items/any
278                                 "type":
279                                     ↪ "integer",
280                                 "title": "The
281                                     ↪ pos_y
282                                     ↪ schema",
283                                 "description":
284                                     ↪ "An
285                                     ↪ explanation
286                                     ↪ about the
287                                     ↪ purpose of
288                                     ↪ this
289                                     ↪ instance.",
290                                 "default": 0,
291                                 "examples": [
292                                     150
293                                 ]
294                             }
295                         }
296                     ],
297                     "$id": "The edges schema",
298                     "type": "array",
299                     "$id": "#/properties/nodes/items/anyOf/0/properties/edges",
300                     ↪ "#/properties/nodes/items/anyOf/0/properties/edges"

```

```

301     "description": "An explanation about the purpose of this
      ↪ instance.",
302     "default": [],
303     "examples": [
304         [
305             {
306                 "source": "act-1",
307                 "target": "act-2",
308                 "label": ""
309             }
310         ]
311     ],
312     "additionalItems": true,
313     "items": {
314         "anyOf": [
315             {
316                 "$id": "#/properties/edges/items/anyOf/0",
317                 "type": "object",
318                 "title": "The first anyOf schema",
319                 "description": "An explanation about the
      ↪ purpose of this instance.",
320                 "default": {},
321                 "examples": [
322                     {
323                         "source": "act-1",
324                         "target": "act-2",
325                         "label": ""
326                     }
327                 ],
328                 "required": [
329                     "source",
330                     "target",
331                     "label"
332                 ],
333                 "additionalProperties": true,
334                 "properties": {
335                     "source": {
336                         "$id":
      ↪ "#/properties/edges/items/anyOf/0/properties
337                         "type": "string",

```

```

338     "title": "The source schema",
339     "description": "An explanation about
    ↪ the purpose of this instance.",
340     "default": "",
341     "examples": [
342         "act-1"
343     ]
344 },
345 "target": {
346     "$id":
    ↪ "#/properties/edges/items/anyOf/0/properties/tar
347     "type": "string",
348     "title": "The target schema",
349     "description": "An explanation about
    ↪ the purpose of this instance.",
350     "default": "",
351     "examples": [
352         "act-2"
353     ]
354 },
355 "label": {
356     "$id":
    ↪ "#/properties/edges/items/anyOf/0/properties/labe
357     "type": "string",
358     "title": "The label schema",
359     "description": "An explanation about
    ↪ the purpose of this instance.",
360     "default": "",
361     "examples": [
362         ""
363     ]
364 }
365 }
366 }
367 ],
368 "$id": "#/properties/edges/items"
369 }
370 }
371 }
372 }

```

B.3.2 Activity

```
1  {
2      "$schema": "http://json-schema.org/draft-07/schema",
3      "$id": "http://example.com/example.json",
4      "type": "object",
5      "title": "The root schema",
6      "description": "The root schema comprises the entire JSON
7      ↪ document.",
8      "default": {},
9      "examples": [
10         {
11             "_id": "5ef8157c82d078e2703fdc82",
12             "name": "Form Activity",
13             "properties": [
14                 {
15                     "icon": "http://tiagoc.xyz/files/form.png",
16                     "description": "An online form"
17                 }
18             ],
19             "config_url": "http://tiagoc.xyz/files/form.html",
20             "json_params":
21             ↪ "http://tiagoc.xyz/files/config_fields.json",
22             "user_url": "http://tiagoc.xyz/files/student.html",
23             "analytics": "http://tiagoc.xyz/files/analytics_2.json"
24         }
25     ],
26     "required": [
27         "_id",
28         "name",
29         "properties",
30         "config_url",
31         "json_params",
32         "user_url",
33         "analytics"
34     ],
35     "additionalProperties": true,
36     "properties": {
```

```

35     "_id": {
36         "$id": "#/properties/_id",
37         "type": "string",
38         "title": "The _id schema",
39         "description": "An explanation about the purpose of this
    ↪ instance.",
40         "default": "",
41         "examples": [
42             "5ef8157c82d078e2703fdc82"
43         ]
44     },
45     "name": {
46         "$id": "#/properties/name",
47         "type": "string",
48         "title": "The name schema",
49         "description": "An explanation about the purpose of this
    ↪ instance.",
50         "default": "",
51         "examples": [
52             "Form Activity"
53         ]
54     },
55     "properties": {
56         "$id": "#/properties/properties",
57         "type": "array",
58         "title": "The properties schema",
59         "description": "An explanation about the purpose of this
    ↪ instance.",
60         "default": [],
61         "examples": [
62             [
63                 {
64                     "icon": "http://tiagoc.xyz/files/form.png",
65                     "description": "An online form"
66                 }
67             ]
68         ],
69         "additionalItems": true,
70         "items": {
71             "anyOf": [

```

```

72     {
73         "$id":
74         ↪ "#/properties/properties/items/anyOf/0",
75         "type": "object",
76         "title": "The first anyOf schema",
77         "description": "An explanation about the
78         ↪ purpose of this instance.",
79         "default": {},
80         "examples": [
81             {
82                 "icon":
83                 ↪ "http://tiagoc.xyz/files/form.png",
84                 "description": "An online form"
85             }
86         ],
87         "required": [
88             "icon",
89             "description"
90         ],
91         "additionalProperties": true,
92         "properties": {
93             "icon": {
94                 "$id":
95                 ↪ "#/properties/properties/items/anyOf/0/prop
96                 "type": "string",
97                 "title": "The icon schema",
98                 "description": "An explanation about
99                 ↪ the purpose of this instance.",
100                 "default": "",
101                 "examples": [
102                     ↪ "http://tiagoc.xyz/files/form.png"
103                 ]
104             },
105             "description": {
106                 "$id":
107                 ↪ "#/properties/properties/items/anyOf/0/prop
108                 "type": "string",
109                 "title": "The description schema",

```

```

104         "description": "An explanation about
           ↳ the purpose of this instance.",
105         "default": "",
106         "examples": [
107             "An online form"
108         ]
109     }
110 }
111 }
112 ],
113     "$id": "#/properties/properties/items"
114 }
115 },
116 "config_url": {
117     "$id": "#/properties/config_url",
118     "type": "string",
119     "title": "The config_url schema",
120     "description": "An explanation about the purpose of this
           ↳ instance.",
121     "default": "",
122     "examples": [
123         "http://tiagoc.xyz/files/form.html"
124     ]
125 },
126 "json_params": {
127     "$id": "#/properties/json_params",
128     "type": "string",
129     "title": "The json_params schema",
130     "description": "An explanation about the purpose of this
           ↳ instance.",
131     "default": "",
132     "examples": [
133         "http://tiagoc.xyz/files/config_fields.json"
134     ]
135 },
136 "user_url": {
137     "$id": "#/properties/user_url",
138     "type": "string",
139     "title": "The user_url schema",

```



```
140         "description": "An explanation about the purpose of this
           ↳ instance.",
141         "default": "",
142         "examples": [
143             "http://tiagoc.xyz/files/student.html"
144         ]
145     },
146     "analytics": {
147         "$id": "#/properties/analytics",
148         "type": "string",
149         "title": "The analytics schema",
150         "description": "An explanation about the purpose of this
           ↳ instance.",
151         "default": "",
152         "examples": [
153             "http://tiagoc.xyz/files/analytics_2.json"
154         ]
155     }
156 }
157 }
```

References

- [1] Authoring Game-Based Learning Activities that are Manageable by Teachers.
- [2] Gamification Software Award Winner. Last accessed on July 2, 2020 from <https://gamificationawards.org/blog/gamification-software-award-winner>.
- [3] moleculer vs-seneca-vs-devis | npm trends. Last accessed on July 2, 2020 from <https://www.npmtrends.com/moleculer-vs-seneca-vs-devis>.
- [4] React Bootstrap. Last accessed on July 2, 2020 from <https://react-bootstrap.github.io/>.
- [5] react-cytoscape. Last accessed on July 2, 2020 from <https://www.npmjs.com/package/react-cytoscape>.
- [6] Stack Overflow Developer Survey 2018. Last accessed on July 2, 2020 from https://insights.stackoverflow.com/survey/2018/?utm_source=so-owned&utm_medium=social&utm_campaign=dev-survey-2018&utm_content=social-share.
- [7] Twine / An open-source tool for telling interactive, nonlinear stories. Last accessed on July 2, 2020 from <https://twinery.org/>.
- [8] twine2:how_to_choose_a_story_format [Twine Wiki]. Last accessed on July 2, 2020 from https://twinery.org/wiki/twine2:how_to_choose_a_story_format.
- [9] What are Progressive Web Apps? Last accessed on July 2, 2020 from <https://web.dev/what-are-pwas/>.
- [10] What is Moleculer? Last accessed on July 2, 2020 from <https://moleculer.services/docs/0.14/index.html>.
- [11] What is NoSQL? NoSQL Databases Explained. Last accessed on July 2, 2020 from <https://www.mongodb.com/nosql-explained>.
- [12] What is Padlet? Last accessed on July 2, 2020 from <https://padlet.com>.
- [13] Sakinah S. J. Alhadad, Kate Thompson, Simon Knight, Melinda Lewis, and Jason M. Lodge. Analytics-enabled teaching as design: reconceptualisation and call for research. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, pages 427–435, Sydney, New South Wales, Australia, March 2018. Association for Computing Machinery.
- [14] Ricardo José Baptista, António Coelho, and C. V. de Carvalho. Relation Between Game Genres and Competences for In-Game Certification. 2016. Accepted: 2017-12-31T15:29:22Z.

- [15] Doug Clow. The learning analytics cycle: closing the loop effectively. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, LAK '12, pages 134–138, Vancouver, British Columbia, Canada, April 2012. Association for Computing Machinery.
- [16] Robert Costello. Evaluating E-Learning from an End User Perspective. pages 259–283. July 2013.
- [17] Duarte Cota. Final Project Report, BEng Informatics Engineering. Technical report, Universidade Aberta, Lisbon, Portugal, 2020.
- [18] Cytoscape. What is Cytoscape? Last accessed on July 2, 2020 from https://cytoscape.org/what_is_cytoscape.html#section-cy2.
- [19] Ann Rosnida Md Deni and Zainor Izat Zainal. Padlet as an educational tool: pedagogical considerations and lessons learnt. In *Proceedings of the 10th International Conference on Education Technology and Computers*, ICETC '18, pages 156–162, Tokyo, Japan, October 2018. Association for Computing Machinery.
- [20] Schlemmer *et al.* Inven!RA: the need for a technological platform to support digital transformation in education within Inventive Reticular Atopic ecosystems. UNPUBLISHED.
- [21] Cordis Europa. Breaking Educational Barriers with Contextualised, Pervasive and Gameful Learning (BEACONING) | BEACONING Project | H2020. Last accessed on July 2, 2020 from <https://cordis.europa.eu/project/id/687676>.
- [22] João Miguel Polónia Pascoal Faria. Jogos pervasivo para o ensino. July 2017. Accepted: 2019-02-02T16:47:17Z.
- [23] Rebecca Ferguson. Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5/6):304, 2012.
- [24] Carsten Fortmann-Grote, Jan-Philipp Burchert, Michael Bussmann, Marco Garten, Axel Huebl, Thomas Kluge, and Adrian P. Mancuso. Deliverable D4.4 Simulated coherent scattering data from plasma and non-plasma samples. September 2017. Publisher: Zenodo.
- [25] David Fortunato and Jorge Bernardino. Progressive web apps: An alternative to the native mobile Apps. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6, June 2018.
- [26] Max Franz, Christian T. Lopes, Gerardo Huck, Yue Dong, Onur Sumer, and Gary D. Bader. Cytoscape.js: a graph theory library for visualisation and analysis. *Bioinformatics*, page btv557, September 2015.
- [27] iLRN. About us - The Immersive Learning Research Network, June 2014. Last accessed on July 2, 2020 from <https://immersivelrn.org/about-us/what-is-ilrn>.
- [28] António Coelho João Jacob João Faria Neil Judd Michał Polak Manuel Freire Baltasar Fernández-Manjón Oriol Janés Ivan Martínez Ortiz Antoniu Ștefan Ioana Andreea Ștefan, Massimiliano Cazzaniga. D3.6 System architecture. January 2019. Publisher: Zenodo.
- [29] Mark Otto Jacob Thornton, and Bootstrap. Bootstrap. Last accessed on July 2, 2020 from <https://getbootstrap.com>.

- [30] Kaspersky. Defending digital privacy: taking personal protection to the next level. Last accessed on July 2, 2020 from <https://www.kaspersky.com/blog/global-privacy-report-2020>.
- [31] Virgínia Kastrup. *A Invenção De Si E Do Mundo. Uma Introdução Do Tempo E Do Coletivo No Estudo Da Cognição*. Autêntica, 2007.
- [32] Chris Klimas. klembot/twinejs, June 2020. Last accessed on July 2, 2020 from <https://github.com/klembot/twinejs>.
- [33] MongoDB. The most popular database for modern apps. Last accessed on July 2, 2020 from <https://www.mongodb.com>.
- [34] Luca Morini, Sylvester Arnab, and Michael Loizou. BEACONING D4.8 Gamified Lesson Plans. May 2018. Publisher: Zenodo.
- [35] Node.js. About. Last accessed on July 2, 2020 from <https://nodejs.org/en/about>.
- [36] David Otasek, John H. Morris, Jorge Bouças, Alexander R. Pico, and Barry Demchak. Cytoscape Automation: empowering workflow-based network analysis. *Genome Biology*, 20(1):185, December 2019.
- [37] Ángel Serrano-Laguna, Javier Torrente, Pablo Moreno-Ger, and Baltasar Fernández-Manjón. Application of Learning Analytics in educational videogames. *Entertainment Computing*, 5(4):313–322, December 2014.
- [38] Twitter. React – A JavaScript library for building user interfaces. Last accessed on July 2, 2020 from <https://reactjs.org>.
- [39] Unisinos. Internationalization plan - unisinos. Last accessed on July 2, 2020 from <http://www.unisinos.br/global/en/print/internationalizatio-plan>.