Computer Adaptive Testing for an Intelligent E-Learning Platform

Julián Ricardo Mahecha D’Maria
Computer and Systems Engineer
Code: 300559

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas e Industrial
Bogotá D.C.
Junio de 2014
Computer Adaptive Testing for an Intelligent E-Learning Platform

Julián Ricardo Mahecha D’Maria
Computer and Systems Engineer
Code: 300559

Thesis Work to Obtain the Degree of Magister in Computer and Systems Engineering

Advisor
Jonatan Gomez Perdomo, Ph.D.
Ph.D. in Mathematics, Computer Science Concentration

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas e Industrial
Bogotá D.C.
Junio de 2014
Title in English

Título en español
Pruebas Adaptativas por Computador para una Plataforma Inteligente de Aprendizaje Virtual

Abstract: This Thesis proposes a Computer Adaptive Test model based on Item Response Theory for an E-Learning platform. The Computer Programming course of Universidad Nacional de Colombia is used to proof the concept of this platform. In order to integrate the model, two modules that allow the creation and administration of tests are developed: test management module and test taking module. Every Computer Adaptive Test, uses an item bank to select items according to one of the Item Response Theory models available in the platform (One, Two and Three Parametric Logistic models). Each item bank is created using the experts approach, with the collaboration of professors of the course. The development of the Computer Adaptive Test model is described, including, its versions, trials, and how it is improved using feedback from students and professors. In addition to this, two data mining techniques are used in order to establish users profiles and question associations. These processes uses data that is generated after each trials and uses the behavior of students (their answers) in order to generate useful information for the course. The first data mining process, generates three cluster of students per test using the k-means algorithm, these clusters are analyzed in order to generate a proper description. The resulting descriptions shows that students are group according to its ability level. In the second data mining process the associations rules process generates relation between questions using the Apriori algorithm. There relations are analyzed and reveals relations between topics of the course. This information can be used to review the structure of the course content based on students performance.

Resumen: Esta Tesis propone un modelo de Pruebas Adaptativas por Computador basado en la Teoría de Respuesta al Ítem para una Plataforma Inteligente de Aprendizaje Virtual. El curso de Programación de Computadores de la Universidad Nacional de Colombia se utiliza para probar el concepto de esta plataforma. Con el fin de integrar el modelo, dos módulos que permiten la creación y administración de las pruebas se desarrollan en la plataforma: el módulo de gestión de pruebas y el modulo de presentación de prueba. Cada prueba adaptativa, utiliza un banco de ítems para seleccionar las preguntas de acuerdo a uno de los modelos de Teoría de Respuesta al Ítem que están disponibles en la plataforma (One, Two or Three Parametric Logistic models). Cada banco de ítems se crea utilizando el enfoque basado en expertos, con la colaboración de los profesores del curso. El desarrollo del modelo de Pruebas Adaptativas por Computador es descrito, incluyendo las versiones del modelo, las pruebas realizadas y las mejoras propuestas que utilizan la retroalimentación de los estudiantes y profesores. Además de esto, dos técnicas de minería de datos se utilizan para establecer perfiles de los usuarios y las asociaciones de las preguntas. Estos procesos utilizan datos que se generan después de cada prueba y utilizan el comportamiento de los estudiantes (sus respuestas) con el fin de generar información útil para el curso. El primer proceso de minería de datos, genera tres grupos de estudiantes por cada prueba, utilizando el algoritmo de k-medias, estas agrupaciones son analizados con el fin de generar una descripción apropiada. Las descripciones resultantes
muestran que los estudiantes son agrupados de acuerdo a su nivel de habilidad. En el segundo proceso de minería de datos, el proceso de reglas de asociacion genera relaciones entre las preguntas utilizando el algoritmo Apriori. Se analizan estas relaciones y esto revela relaciones entre los temas del curso. Esta información puede ser usada para revisar la estructura de los contenidos del curso con base en el desempeño de los estudiantes.

**Keywords:** E-Learning, Virtual Platform, Computer Adaptive Tests, Item Response Theory, Data mining.

**Palabras clave:** Aprendizaje en Línea, Plataforma Virtual, Pruebas Adaptativas por Computador, Teoría de Respuesta al Item, Minería de Datos.
Dedication

This Thesis is dedicated to my family: my mom, my aunt and my brother who always offered me unconditional love and support. And especially to Diana Garcia, who in the last years was always there with me when I needed the most and always knew how to make me feel better.
Acknowledgements

I would like to thank to my Thesis advisor Jonatan Gómez Perdomo for his dedication, direction, and discussion. Without him the goals for this Thesis would not have been achieved.

Additionally, I would like to thank to the people of the Alife research group from Universidad Nacional de Colombia. Without their time and cooperation, this work not have been possible to be completed.

This work is a subproject of the project “Plataforma Inteligente de Aprendizaje Virtual para el Desarrollo de Programas Académicos Virtuales: Caso Concreto Curso de Programación de Computadores” supported by Colciencias. Grant: 110152128971.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>I</td>
</tr>
<tr>
<td>List of Tables</td>
<td>III</td>
</tr>
<tr>
<td>List of Figures</td>
<td>IV</td>
</tr>
<tr>
<td>Introduction</td>
<td>VI</td>
</tr>
<tr>
<td>1. Background</td>
<td>1</td>
</tr>
<tr>
<td>1.1 E-Learning</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Computer Adaptive Tests</td>
<td>3</td>
</tr>
<tr>
<td>1.2.1 Item Selection</td>
<td>4</td>
</tr>
<tr>
<td>1.2.1.1 One Parametric Logistic Model</td>
<td>5</td>
</tr>
<tr>
<td>1.2.1.2 Two Parametric Logistic Model (2PL)</td>
<td>6</td>
</tr>
<tr>
<td>1.2.1.3 Three Parametric Logistic Model (3PL)</td>
<td>8</td>
</tr>
<tr>
<td>1.2.2 Stopping Rules</td>
<td>9</td>
</tr>
<tr>
<td>1.2.3 Item Calibration</td>
<td>10</td>
</tr>
<tr>
<td>1.3 Adaptability in E-Learning</td>
<td>10</td>
</tr>
<tr>
<td>1.4 E-Learning in Colombia</td>
<td>13</td>
</tr>
<tr>
<td>1.5 Data Mining in E-Learning</td>
<td>14</td>
</tr>
<tr>
<td>1.5.1 K-Means Algorithm</td>
<td>18</td>
</tr>
<tr>
<td>1.5.1.1 Silhouette Coefficient</td>
<td>18</td>
</tr>
<tr>
<td>1.5.1.2 Davies Bouldin Index</td>
<td>19</td>
</tr>
<tr>
<td>1.5.2 Association Rules</td>
<td>19</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>20</td>
</tr>
</tbody>
</table>
CONTENTS

II

2.2 Item Creation ................................................. 22
2.3 Basic Model .................................................. 25
2.4 Question History Model .................................... 29
2.5 Topic History Model ....................................... 32
2.6 Summary ...................................................... 35

3. Knowledge Discovery after Tests 37

3.1 Introduction .................................................. 37
3.2 Data Base Model ............................................. 38
3.3 User Profiling ............................................... 38
  3.3.1 Data Selection ............................................ 38
  3.3.2 Pre-Processing and Transformation ...................... 42
  3.3.3 Data Mining - Clustering ................................ 42
  3.3.4 Interpretation and Evaluation .......................... 45
3.4 Question Associations ....................................... 50
  3.4.1 Data Selection, Pre-Processing and Transformation .... 50
  3.4.2 Data Mining - Association ............................... 50
  3.4.3 Interpretation and Evaluation .......................... 51
3.5 Summary ...................................................... 54

Conclusions ..................................................... 55

Future Work ...................................................... 57

Bibliography ..................................................... 58
## List of Tables

1.1 X → Supported, * → Planned to be supported, ** → not part of the original system, but added as an extension .......................... 11

1.2 E-Learning systems that use data mining .......................... 17

2.1 Scales of IRT parameter in the questions interfaces and internally used in IRT models .......................................................... 25

2.2 General grades of first test (2013-I) ........................................ 27

2.3 General grades of second test (2013-I) ................................. 32

2.4 General grades of first test (2013-II) ..................................... 34

2.5 General grades of second test (2013-II) ............................... 34

2.6 General grades of third test (2013-II) ................................... 35

3.1 Total students per test ........................................................... 39

3.2 Total questions taken per test .............................................. 40

3.3 Example of vectors from Pre-Processing and Transformation ........ 42

3.4 Example of vectors comparison ........................................... 43

3.5 Example vectors per difficulty ............................................ 44

3.6 Summary of cluster description ........................................... 49

3.7 Students in tests #1 and #2 ............................................... 49

3.8 Students in test #3 and #4 ................................................ 50

3.9 Students in test #4 and #5 ................................................ 50

3.10 Example vector for correct questions .................................. 51

3.11 Example vector for incorrect questions ............................... 51

3.12 Summary of association rules for all five tests ....................... 52

3.13 Summary of association by topics ..................................... 53
# List of Figures

1.1 Answer Probability of a Dichotomous Question \[51\] ...................... 5  
1.2 Item Response Function of 1PL \[51\] ................................. 5  
1.3 Example of likelihood function with at least one right answer and one wrong answer \[37\]  ............................................................... 7  
1.4 Example of likelihood function with all answers right \[37\] ..... 7  
1.5 Item Response Function of 2PL, introducing discrimination parameter \[51\] 7  
1.6 Item Response Function of 3PL, introducing guessing parameter \[51\] 8  
1.7 Features of E-Learning Systems, taken from \[24\] ................. 14  
1.8 Web mining structure \[24\] ............................................ 15  

2.1 Computer Adaptive Test process \[38\] .................................. 21  
2.2 Questions Interface ..................................................... 23  
2.3 Bags Interface .......................................................... 23  
2.4 Test Interface ............................................................ 24  
2.5 Basic Model structure ................................................. 26  
2.6 General performance of first test (2013-I) .......................... 27  
2.7 Progress of a student in the first test (I) ........................... 28  
2.8 Progress of a student in the first test (II) .......................... 28  
2.9 Progress of a student in the first test (III) ......................... 29  
2.10 Progress of a student in the first test (IV) ...................... 29  
2.11 Progress of a student in the first test (V) ....................... 30  
2.12 Question History Model ............................................. 31  
2.13 Progress of a student in the second test ......................... 31  
2.14 General performance of second test (2013-I) .................... 32  
2.15 Topic History Model ................................................. 33  
2.16 General performance of first test (2013-II) ...................... 34
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.17</td>
<td>General performance of second test (2013-II)</td>
<td>34</td>
</tr>
<tr>
<td>2.18</td>
<td>General performance of third test (2013-II)</td>
<td>35</td>
</tr>
<tr>
<td>3.1</td>
<td>Database schema for the evaluation module</td>
<td>39</td>
</tr>
<tr>
<td>3.2</td>
<td>Tables use for data selection</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>Top ten correct and incorrect questions per test, (a-b) for test #1, (c-d) for test #2, (e-f) for test #3, (g-h) for test #4 and (i-j) for test #5</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>Silhouette Coefficient for Test 1</td>
<td>45</td>
</tr>
<tr>
<td>3.5</td>
<td>Davies Bouldin Index for Test 1</td>
<td>45</td>
</tr>
<tr>
<td>3.6</td>
<td>Students grouping in Test 1</td>
<td>46</td>
</tr>
<tr>
<td>3.7</td>
<td>Distribution of correct and incorrect questions per cluster</td>
<td>46</td>
</tr>
<tr>
<td>3.8</td>
<td>Distribution of correct and incorrect questions with difficulty per cluster</td>
<td>46</td>
</tr>
<tr>
<td>3.9</td>
<td>Analysis Clusters Test 1, (A) First Answer, (B) First Difficulty</td>
<td>47</td>
</tr>
<tr>
<td>3.10</td>
<td>Question #357</td>
<td>53</td>
</tr>
<tr>
<td>3.11</td>
<td>Question #281</td>
<td>54</td>
</tr>
</tbody>
</table>
Introduction

People learn new things every day, while they are at home, at work, or at any other place or situation. In general, the learning process is defined as the process of acquiring new knowledge, abilities and skills and how the existing knowledge is modified in order to solve problems [32]. However, not all people learn the same features of a particular situation. Given that everyone learns in a different way, Kahiigi et al. [32] classify learners in the following three categories: visual learners, auditory learners, and tactile/kinaesthetic learners. The learning process of visual learners is better when information has visual content such as images, demonstrations, facial and body language of the teacher, etc. For auditory learners, their appropriation of knowledge is better when they listen to the information; for instance: verbal lectures, discussions, talking things and listening to others. In the case of tactile/kinaesthetic learners, they prefer to interact with the information by experiencing, reflecting, and doing things. These learners would rather manipulate real objects or simulated environments.

Another important aspect of the learning process is the teacher-student relationship. This relation includes how teachers and students interact and the way teachers present information to students. Considering the teacher-student relation, Kahiigi et al. [32] define the following five learning methods:

- **Traditional Learning**: is also known as face-to-face sessions, in this type of learning teachers and students are located in the same place (classroom), and the teacher gives course material and content to students. This is a teacher centered method, i.e. the teacher provides information to students. Preparation for assessments (evaluations, tests, etc.) is limited to student notes, limiting the knowledge acquisition boundary.

- **E-Learning**: is the use of information and communication technologies in order to support the learning process ubiquitously.

- **Blended Learning**: is the combination of several learning methods including face-to-face interactions, classroom activities, live E-learning and self-learning, in order to maximize the student’s learning potential.

- **Mobile Learning**: is the use of portable technologies such as mobile phones, PDAs or tablets in order to deliver content to students.

- **Personalized Learning**: is specially focused in supporting and facilitating individual learning. This method considers that each student learns in his or her own way; it also attends to students needs and interests in a productive way.
In particular, the E-Learning concept has evolved during the past decade, this is a **distance learning** method which integrates information technologies with pedagogical elements in order to teach and train students in a particular topic, independent of his or her location, or his or her time availability (nights, hours, days) [20, 56, 33, 8, 32].

E-Learning is a priority for the Government of Colombia. It is one of the strategic areas of **Eje Educación del Plan Nacional de Desarrollo en TICs** (Education Axis of National Development Plan ICTs), and it is also an active research topic [61, 16, 66, 3, 34]. In Colombia, some institutions are working on the E-Learning area. Institutions such as SENA (Servicio Nacional de Aprendizaje; in english: National Learning Service) [61], Universidad Autónoma de Bucaramanga-UNAB Virtual [34], Universidad Nacional Abierta y a Distancia [66], Universidad Tecnológica de Pereira, -with its Miriada X platform [15], which is the first MOOC made in Colombia- and the Universidad Nacional de Colombia have several projects [16, 67]. Most of these institutions use systems such as Blackboard [29] or Moodle [1] to offer courses.

However, E-Learning in Colombia remains to be quite basic. This is mainly due to the development and use of E-Learning platforms that are focused on Learning Management Systems [9, 5, 17] and Content Management Systems [25, 4]. These approaches do not include some essential components in the student’s learning process, such as virtual tutors (Intelligent Tutoring Systems) [53, 46, 39], adaptive evaluations (Computer Adaptive Test) [58, 2, 62] and interactive/simulation tools [7, 59, 31]. The use of these components change the learning process from a knowledge acquisition model (passive-receiver model) to one where the interaction and learning are guided by the platform (active-receiver model).

In this direction, an Intelligent E-Learning Platform is being developed at Universidad Nacional de Colombia. The computer programming course has been used to proof the concept of the platform [24, 23]. This platform presents the course content like a map, which describes the dependencies between topics and allows the students to navigate among them in any order. This platform offers the topics content in a “lectures schema”, which is based on a series of materials that include videos, documents and exercises created by the teacher; additionally, each class has a chat-room where students can discuss about the topic. The platform takes into account the categories of students defined by Kahiigi et al. [32]. The use of multimedia material (videos) improves the learning process of visual and auditory learners; meanwhile, the learning process for tactile/kinaesthetic learners is enhanced by providing them with interaction tools such as the chat-room within each class and also by giving them exercises to practice concepts of the class.

In particular, the Intelligent E-Learning Platform supports the learning process of students from different academic programs of the Universidad Nacional de Colombia. For instance, courses like the computer programming course, have students from engineering (that includes more than 5 different degree programs), economy, statistics and mathematics, among others. This situation gives a great diversity of students to these courses considering that each one has its own background. This diversity makes the evaluation process a hard task. The platform requires an evaluation methodology/strategy that assesses a diversity of students properly and assigns a grade according to the student capabilities.

The Computer Adaptive Tests (CATs) strategy supports the measurement process of student’s knowledge. Computer Adaptive Test is an evaluation method which considers the individual progress of each student through a test to measure his or her knowledge and grade it. The CATs strategy involves three main processes: test elaboration, test taking
and test grading \cite{38}. Test elaboration includes the definition of an item bank. Each item has some parameters that are used when taking the test. In the test taking process the system selects an initial question and presents it to a student, he or she answers the question and the system selects the next one. CATs uses Item Response Theory (IRT) models in order to select questions for a test. IRT is based on the proposition that the probability of answer a question is a mathematical function that depends on the item parameters. IRT models estimate the student ability (knowledge level) and use it to select the next question. For example, if a student answered a question correctly, his or her ability is higher than the question level, so the next question will be more challenging, but if he or she answers incorrectly, the next one will be easier \cite{37}. Finally, the test grading process uses the last estimated ability and adjusts it to a proper grade scale (e.g 0-5 scale).

Goal

In this Thesis, a Computer Adaptive Test model based on Item Response Theory for an E-Learning platform is proposed. The Computer Programming course of Universidad Nacional de Colombia is used to proof the concept of the platform.

This Thesis shows the building process of a Computer Adaptive Test model from scratch, including: the creation of an item bank, the construction and administration of exams, the definition of user profiling and the discovering of associations between questions.

The purpose of this work is to provide a Computer Adaptive Test model based on Item Response Theory, data mining and social behaviour and integrate them into an E-Learning platform. Specifically, this research concentrates on:

- **Defining a Computer Adaptive Tests model based on Item Response Theory.** To propose a CAT model, the first task is to identify key features of CATs, and then, to analyze the IRT and identify its models and key concepts. Finally, the IRT concepts and models are combined with the CAT features in order to create a complete model.

- **Establishing a model of Computer Adaptive Tests based on data mining.** To define a CAT model based on data mining, the first step is to review data mining techniques that are used in E-Learning and CATs, in order to identify possible approaches for the model. When some techniques are identified, the data, available after the exams, is analyzed in order to discover relations between students and questions. Using this data, some datasets that represent the relation between students and questions are generated. These datasets are used in a clustering process that creates student’s clusters and that represents users profiles. Each cluster is analyzed in order to assign to it a proper description.

- **Determining a model of Computer Adaptive Tests based on social behavior.** To establish a CAT model based on social behavior, the first task is to create select data that relates questions with students. Then, use the data to create two datasets per test, one that relates students and correct answers and another one that relates students and incorrect answers. These datasets are used to determine hidden relations between questions and can reveal valuable information for the course. Such
relations are represented by association rules. These association rules are analyzed to find out its meaning and its relevance for the course.

- **Integrating, deployment and evaluating the model with the students and professors of the computer programming course.** To integrate the CAT mode into the Intelligent E-Learning Platform is required to develop two modules: the test management module and the test taking module. The test management module is used to create the item bank for each test. Additionally, this module allows test creation and test configuration. The test taking module is used by students to take tests; this module is in charge of question selection and grading. The computer programming course is used to proof the concept of the CAT model. The CAT model is improved by using feedback from some of the trials.

**Main Contributions**

The main contribution of this work is to develop a Computer Adaptive Test model based on Item Response Theory. The following is a list of this Thesis contributions:

**Computer Adaptive Test model integrated to an E-Learning Platform**

A Computer Adaptive Test model based on Item Response Theory is designed and it is integrated into the Intelligent E-Learning Platform. An article called “Evolution of Teaching and Evaluation Methodologies: The Experience in the Computer Programming Course at Universidad Nacional de Colombia” presents the general process of the model (this work is approved to be published in the *Revista de Ingeniería e Investigación de la Universidad Nacional de Colombia* (Journal of Engineering Research of Universidad Nacional de Colombia)[22]. The model includes: item selection, ability estimation and stopping rules. The item selection process supports three IRT models: One, Two or Three Parametric Logistic models (1PL, 2PL and 3PL). The model uses a maximum likelihood function to make the ability estimation of students. Additionally, the model includes two types of stopping rules: fixed length and variable length options. Two modules are created in the E-Learning platform; these modules manage the creation and the taking of tests. The test management module is used by professors and coordinators; it allows the elaboration of questions (content, answers options and the setup of IRT parameters) and the test creation using these questions. Moreover, the creation module allows to configure tests (by selecting a Item Response Theory model and a stopping rule) and to schedule it in a specific date. The test taking module is where students take tests; this module compute the ability estimation, the item selection and the grading process. The platform development is published in two articles, the first one entitled “A Didactic E-Learning Platform with Open Content Navigation and Adaptive Exercises” in the proceedings of the International Conference on Education and E-Learning Innovations (ICEELI) [24], and the second one entitled “Plataforma de Aprendizaje y Cursos Masivos Abiertos en Línea de la Universidad Nacional de Colombia” (in english, “Learning Platform and Massive Open Online Courses at Universidad Nacional de Colombia”) in the proceedings of Virtual Educa 2013 [23].
Item Banks for the Computer Programming Course

Five Item Banks for the Computer Programming Course are created. These Banks include over 950 questions; each question is setup with three parameters: difficulty, discrimination and guessing. The five banks correspond to five tests made in 2013. Normally, the course applies three tests in a regular academic term; however, in the 2013-I period only two tests are made. Questions of those banks are a base for future tests of the course. Additionally, other courses can take these questions as an example of how to create an item bank from scratch. Those banks are used to make five trials of the Computer Adaptive Test model. During these trials, the setup of parameters, the relation between them and how they affect the performance and grade of students are studied. Trials are analyzed and the model is improved based on the analysis of results.

Data Mining model in order to create users profiles and question associations

The design and implementation of a data mining model to create users profiles and question associations is created. The model has two main processes: user profiles and discovering of question associations. Datasets that relate students with questions are created, one dataset per test. Moreover, in order to compare vectors within datasets, a dissimilarity measure is proposed. To create users profiles, the K-means algorithm is used along with the proposed comparison metric over the generated datasets. The resulting clusters are evaluated using the silhouette coefficient and Davies Bouldin index. In order to find the more suitable \( k \) an exploration process is done. Once the \( k \) is selected, labels (or descriptions) for each cluster are defined. For discovering question associations, two new datasets are created, one for correct questions and one for incorrect ones. The \textit{Apriori} algorithm is used to discover relations between questions. An association set is generated to each dataset and it is analyzed. The associations reveal relations between question topics.

Thesis Outline

The structure of this Thesis is as follows:

- **Chapter 1** provides an augmented description of the background about E-Learning, Computer Adaptive Tests and Data Mining in E-Learning. Additionally, the basic concepts applied in the development of the Thesis are presented.

- **Chapter 2** presents the creation process of the Computer Adaptive Test model, some trials of the model, and their results. On addition to this the analysis of these trials and the feedback given by students and professors is presented. This feedback is used to improve the model and to validate it again.

- **Chapter 3** presents the Data Mining process for user profiling and discovering of question associations. This process includes: the generation of the datasets, their pre-processing, the application of data mining techniques, the generated results and the analysis and interpretation of these results.

- **Chapter 4** draws some conclusions and future work.
In the last three decades (thanks to Computing and Communications technologies), educational models have been transformed from monolithic linear models to flexible nonlinear models centered in the skills and abilities of students [20, 56, 33, 8]. These flexible and nonlinear models allow students to learn in their own way and at their own pace, guided by some online tools, independent of location (city, region), connection conditions (local or internet) or student time. [8, 20]. This transformation includes educational processes from classroom-based education to distance learning (using radio, TV, internet and mobile devices) [68].

1.1 E-Learning

E-Learning is a distance educational method which integrates information technologies with pedagogical elements in order to teach and train students in a particular topic [20, 56, 33, 8]. Garrison and Anderson [20] defines E-Learning as “electronically mediated asynchronous and synchronous communication for the purpose of constructing and confirming knowledge”. At mid-1990s, when the World Wide Web was advancing and growing fast, the academic community was interested in creating asynchronous discussion groups. The E-Learning’s goal is to create a research community that is independent of time and location, through the use of communication technologies and information.

Studies developed by Garrison and Anderson [20], Salmon [56], Kearskey [33], Bates [8] and Moreno et al. [67], can be highlighted in the E-Learning area. Garrison and Anderson [20] present E-Learning systems from a conceptual point of view. They present the theoretical foundations, including fundamental concepts of education, theoretical principles and how to apply them. Additionally, they present learning communities and explain three important factors: social presence, cognitive presence and teaching presence. They finally show an overview of the different technologies that have been used in education, and the newest technologies that allow E-Learning.

According to Salmon [56], E-Learning can be characterized by two fundamental elements: the definition of appropriate learning activities and the support of the teacher in the learning process.
A discussion of online education elements (E-Learning and teaching) is presented by Kearsley [33]. He explains the basic concepts of online education, the design and development of online courses (planning, organization, content and politics) and future trends.

Bates [8] presents guidelines of how to choose a technology to be used in the distance learning process. He presents a review of information technologies from print media to websites and its uses in education. Additional to this, he exposes a selection criteria for technologies in distance learning and he remarks the use of web technologies for distance learning.

Moreno et al. [67] describes different components of an E-Learning system, the differences between these components and proposes a classification structure of them by using an educational taxonomy. This taxonomy is divided into four groups: Learning Management Systems, Learning Content Management Systems, Adaptive Learning Systems and Adaptive Educational Hypermedia Systems.

Within E-Learning, there are some strategies that can be used; among them the student centered strategy is found. This strategy is focused on the students, their interests, their abilities, their learning styles, etc. García-Barrios et al. [19], McCombs and Vakili [43], Modritscher and Wild [44], López [40], and Lee and Park [50], have worked in the study of student centered E-Learning systems.

García-Barrios et al. [19] define a generic customization model (based on components) by studying some relevant elements of adaptation in the learning process. They conclude that personalization issues can be solved through a generic personalization model, as well as extended and enhanced through its interpretation as a part of a general adaptation model.

McCombs and Vakili propose a framework for the development of student-centered learning models [43]. This framework is built upon the principles of student-centered psychology and has been validated by several studies made by the American Psychological Association.

López proposes a model of adaptive re-planning activities in virtual courses using concepts of intelligent agents [40]. Basically, the proposal has six intelligent agents responsible for the administration and recommendation of activities (from current management to adaptive assessment).

Modritscher and Wild [44] propose a “bottom-up” student-centered model based on the ideas of personal learning environments, learning activities, student interactions, learning patterns and generating good support practices.

Finally, Lee and Park [50] summarize the characteristics of some elements of an adaptive learning system, and divided it into the following categories: macro-adaptive instructional systems, macro-adaptive instructional models, aptitude-treatment interaction models and micro-adaptive instructional models. They presented some work done in each of these.

In general, an E-Learning system is composed by several subsystems including, but not limited to the following:

- **Learning Management System** [10] [6] [18]: In charge of managing, distributing and controlling training activities. In particular, this subsystem manages users accounts and learning resources (documents, activities, assignments, etc) Additionally,
it follows the learning process of each student, applies tests, generates individual reports, and manages communication resources such as chat, videoconferences and blogs.

- **Content Management System** [25 4]: In charge of defining and maintaining the educational program content (usually courses content). This is done by defining content units that may be managed, personalized and used at different moments of the learning process.

- **Intelligent Tutoring System** [53 46 39]: In charge of guiding students, in an interactive manner, through the educational program content by using feedback, i.e., according to tutor system’s perception about students.

- **Topic interaction and simulation tools**: A set of computational elements such as programming compilers, virtual laboratories, painting tools, CAD (Computer Aided Design) engines, etc., that are integrated in the E-Learning system, in order to allow students to apply or test their learned skills and abilities.

- **Computer Adaptive Test** [58 2 62]: In charge of maintaining and applying adaptive tests to students. An adaptive test is one that is able to select questions according to answers provided by a student. In this way, the test provides a better characterization of the level reached by the student, his or her difficulties and achievements.

### 1.2 Computer Adaptive Tests

Computer Adaptive Test (CAT), as its name implies, is a software application where test content (items) adapts to each individual student. In a CAT, the system estimates the examinee’s knowledge level (based on an initial item response) and adapts according to it, providing an easier item or a harder one. Wainer describes how to build, maintain and operate Computer Adaptive Test systems. Additionally, he explains some subjects to be considered in a CAT system development, including: how to design and operate a CAT system, item groups, item response theory, etc.


Olea et al. [48] present characteristics, advantages and disadvantages of computer-adaptive testing from a psychological point of view. They present an English vocabulary test for Spanish speakers, including both adaptive models and classical models. Additionally, they compare the effects of computer-adaptive tests against computer-fixed tests. They discover a significant increase of correct responses, ability estimation, and a decrease of the anxiety level in the adaptive test.

Tzai-hsing et al. [60] apply Item Response Theories along with adaptive testing techniques in order to build an item bank and an adaptive test system on the web. Their results show that when the test manager assigns an appropriate stopping condition and a maximum number of items, the test is short and has a good quality under these conditions.

Jiménez [30] presents an adaptive test model based on Item Response Theory techniques for student knowledge level estimation, applied to intelligent tutors systems. She
compares artificial intelligence techniques that are used in online course evaluation. Her model is tested with simulated students. She mentions that one challenge in CAT is the item bank creation.

In general, a CAT includes the following stages: item selection, ability estimation, stopping rules and item calibration.

1.2.1 Item Selection

Generally, item selection is made by using Item Response Theory (IRT). IRT uses the examinee’s answer (correct or incorrect) and the difficulty of the questions (other parameters could be included) in order to estimate the examinee’s ability level and select an item according to it.

The selection of the first item is a particular case of the item selection process. In general, at the beginning of the test, there is a small amount of information of the examinee (in some cases, none). If there is some prior information, e.g. education history, performance in similar subjects or previous CAT scores, the first item could be selected based on that information [37].

In the other hand, if there is no information, the selection has to be done with a different method. In 1980, Lord [41] stated that “unless the test is very short, a poor choice of the first item will have little effect on the final result”. Considering this assumption and the fact that there is no prior information available, a random selection for the first item is a valid option. Nevertheless, random selection has limitations. If the selected item is located at the beginning or at the end of the difficulty scale (very easy or very hard), the examinee could experience psychological reactions such as underestimation of the test or panic. Additionally, items of the end of the scale are less useful, because at the beginning of the test, they give less information of the examinee than the questions in the middle of the scale [37].

In the work presented by Lilley [37], difficulty values between -3 and 3 are used; she selects the first item of the test as the one whose difficulty is the closest to 0 (the middle difficulty). Linacre [38] suggests another option which can be used in tests with a pass-fail level criterion. If this criterion is well defined, the first item could be selected as a question with a difficulty slightly below the pass-fail level. In this way, the examinees with an ability close to the pass-fail level are likely to approve, and this will encourage them through the test.

Once the first item is selected and the student has answered it, the selection of the following items is done by using IRT. IRT models represent the probability of answering to an item with a logistic function. Some IRT models work with dichotomous items, i.e. they only have correct or incorrect answer (there are no partially correct answers). The probability functions associated to dichotomous items are shown in Figure 1.1; the red line represents correct answer probability and the black line represents the incorrect answer probability. The first one is represented by \( P(\theta) \), where \( \theta \) is the ability of the examinee. The second one is defined as \( Q(\theta) = 1 - P(\theta) \).

IRT models try to predict the answering probability of a person with an ability level to a specific item. Items differ from each other; for example, some may be easier (or harder) than others. In general, the probability that a examinee answers an item correctly is represented by \( P_{ij} \), where \( i \) refers to the \( i^{th} \) item and \( j \) is the \( j^{th} \) examinee. In IRT,
CHAPTER 1. BACKGROUND

The probability function $P_{ij}$ is called Item Response Function. Each IRT model considers some parameters that are included into $P_{ij}$.

1.2.1.1 One Parametric Logistic Model

The One Parametric Logistic model (1PL) is the simplest in IRT. It includes the ability of the examinee $\theta_j$, the difficulty of items $b_i$ and its Item Response Function. Equation (1.1) shows the Item Response Function used in this model:

$$P_{ij}(\theta_j, b_i) = \frac{1}{1 + e^{(\theta_j - b_i)}}$$  \hspace{1cm} (1.1)

A graphical representation of the Item Response Function is shown in Figure 1.2. The red line shows that an examinee with ability $\theta = 1$, has 50% chance of correctly answering the item with difficulty $b_i$ associated to that function.
Each item in a test provides a small amount of information about the examinee; this information depends on how close the difficulty of the item matches the examinee’s ability. The amount of information is represented by the **Item Information Function**. Equation (1.2) shows the Item Information Function for 1PL model \( I_i(\theta, b_i) \) and represents the amount of information provided by a particular item, \( i \).

\[
I_i(\theta, b_i) = P_i(\theta, b_i)Q_i(\theta, b_i)
\]  
(1.2)

The Item Information Function gives specific information about the examinee. In order to have an idea of the examinee’s ability, various items spread over the possible ability values are needed. In order to obtain a more accurate information about the examinee’s ability, the information of all items in a test is considered. This information is included in the **Test Information Function**, which is the sum of all item information functions in a test. Equation (1.3) show the Test Information Function for 1PL model; \( I(\theta) \) represents the information of a test.

\[
I(\theta) = \sum_i I_i(\theta, b_i)
\]  
(1.3)

In order to measure the accuracy of the test information function, the **Standard Error of Measurement** (SEM) is used. Equation (1.4) shows the SEM of the Test Information Function for 1PL:

\[
SEM(\theta) = \sqrt{\frac{1}{I(\theta)}} = \sqrt{\frac{1}{\sum_i P_i(\theta, b_i)Q_i(\theta, b_i)}}
\]  
(1.4)

To estimate the examinee’s ability, a **Maximum Likelihood Function** is used; it assumes local independence between items. The Maximum Likelihood Function is shown in Equation (1.5), where \( u_i \in (0, 1) \) is the answer of an item \( i \) (0 if it is incorrect and 1 if it is correct). When the function reaches its maximum, the value of \( \theta \) represents the examinee’s ability after answering a series of items.

\[
L(\theta) = \prod_i P_i(\theta, b_i)^{u_i}Q_i(\theta, b_i)^{1-u_i}
\]  
(1.5)

Given that \( P_i \) and \( Q_i \) are logistic functions, it is required that the examinee answers at least one item correctly and one incorrectly in order to estimate the Maximum Likelihood Function. If all answers are correct (or incorrect), the result function has an asymptote and the ability can not be measured properly. Figure 1.3 shows a example of the Maximum Likelihood Function when there is at least one correct and one incorrect answer. Figure 1.4 shows an example of what happen when all the answers are correct, and the ability cannot be estimated.

### 1.2.1.2 Two Parametric Logistic Model (2PL)

The Two Parametric Logistic model, uses the examinee’s ability and two parameters in order to predict the answer probability. This model uses the difficulty parameter and adds a new one represented by \( a_i \), which indicates the discrimination of an item. The \( a_i \)
parameter indicates how much difference the question will represent between an examinee with $\theta = x$ and another one with $\theta = x + \delta$, delta being a small increase.

Figure 1.5 shows the variation of the Item Response Function in the 2PL model compared to the 1PL model. The black line represents the Item Response Function of the 1PL model (where $a_i = 1$). The blue line includes $a_i > 1$; therefore, the curve is steeper than the black one. In green, a function with a higher difficulty and the same discrimination of the black one is presented in order to illustrate the difference between $b_i$ and $a_i$ parameters.

\[
P_{ij}(\theta_j, b_i, a_i) = \frac{1}{1 + e^{a_i(\theta_j - b_j)}}
\]  

Equation (1.6) shows the Item Response Function for the 2PL model:

\[
P_{ij}(\theta_j, b_i, a_i) = \frac{1}{1 + e^{a_i(\theta_j - b_j)}}
\]

Equation (1.7) shows the Item Information Function for the 2PL model:
\[ I_i(\theta, b_i, a_i) = a_i^2 P_i(\theta, b_i, a_i) Q_i(\theta, b_i, a_i) \]  \hspace{1cm} (1.7)

The Test Information Function for the 2PL is shown in Equation (1.8):

\[ I(\theta) = \sum_i I_{ij}(\theta, b_i, a_i) = \sum_i a_i^2 P_i(\theta, b_i, a_i) Q_i(\theta, b_i, a_i) \]  \hspace{1cm} (1.8)

The Standard Error Measurement for the 2PL model is shown in Equation (1.9):

\[ SEM(\theta) = \sqrt{\frac{1}{\sum_i a_i^2 P_i(\theta, b_i, a_i) Q_i(\theta, b_i, a_i)}} \]  \hspace{1cm} (1.9)

The ability estimation maintains the general structure of the Maximum Likelihood Function, but uses \( P_i \) and \( Q_i \) with the discrimination parameter \( a_i \) [1.10].

\[ L(\theta) = \prod_i P_i(\theta, b_i, a_i)^{u_i} Q_i(\theta, b_i, a_i)^{1-u_i} \]  \hspace{1cm} (1.10)

### 1.2.1.3 Three Parametric Logistic Model (3PL)

The Three Parametric Logistic model adds a third parameter to the structure of the 2PL model. This third parameter is guessing; as its name indicates, it represents the probability that an examinee with no knowledge will guess an item; it is denoted by \( c_i \in (0, 1) \).

![Figure 1.6. Item Response Function of 3PL, introducing guessing parameter](image)

Figure 1.6 shows an example of the Item Response Function for the 3PL model, where \( b = 0, a = 1.4 \) and \( c = 0.2 \). The \( c \) value represents the lower asymptote in the function, so that the guessing parameter will move the lower asymptote up or down. In this model, the \( a \) parameter has the same behavior as in the 2PL model.

The 3PL model has the same structure as the 2PL model. It includes the guessing parameter in the following functions: Item Response Function, Item Information Func-
CHAPTER 1. BACKGROUND

The Item Response Function for the 3PL model is shown in Equation (1.11):

\[
P_{ij}(\theta_j, b_i, a_i, c_i) = c_i \frac{1 - c_i}{1 + e^{a_i(\theta_j - b_j)}}
\]

(1.11)

The Item Information Function for the 3PL model is shown in Equation (1.12):

\[
I_i(\theta, b_i, a_i, c_i) = \frac{a^2 Q_i(\theta)}{P_i(\theta)} \left( \frac{P_i(\theta) - c_i}{1 - c_i} \right)^2
\]

(1.12)

In the 3PL model the Test Information Function is shown in Equation (1.13):

\[
I(\theta) = \sum_i I_i(\theta, b_i, a_i, c_i) = \sum_i \frac{a^2 Q_i(\theta)}{P_i(\theta)} \left( \frac{P_i(\theta) - c_i}{1 - c_i} \right)^2
\]

(1.13)

The Standard Error Measurement for this model is shown in Equation (1.14):

\[
SEM(\theta) = \sqrt{\frac{1}{\sum_i a^2 Q_i(\theta) \left( \frac{P_i(\theta) - c_i}{1 - c_i} \right)^2}}
\]

(1.14)

The ability estimation maintains the general structure of Maximum Likelihood Function, but uses \( P_i \) and \( Q_i \) with the guessing parameter. Equation (1.15) represents the function:

\[
L(\theta) = \prod_i P_i(\theta, b_i, a_i, c_i)^{u_i} Q_i(\theta, b_i, a_i, c_i)^{1-u_i}
\]

(1.15)

1.2.2 Stopping Rules

The moment in which the test ends is a very important aspect for CATs. If the test is too short, the ability estimation may be inaccurate, and if it is too long, the examinee can get tired, make careless mistakes and hence lower his or her general performance. Additionally, resources and time may be wasted [37]. In specialized literature, there are two main options of ending a CAT; depending on this, a test can be called a fixed-length test or a variable-length test. A fixed-length test ends when a period of time has passed or when a maximum number of items is presented. In the other hand, a variable-length test uses a metric of an IRT model to stop the test, e.g. the Standard Error Measurement. Some other aspects to be considered in the test finalization are the following [38]:

- **The item bank may run out of questions**: This happens when the item bank is small. For this reason, it is recommended to put as many items as possible in the item bank, at least three times the maximum number of questions of the test.

- **A minimum number of administered items is convenient**: In some cases, if the test stops too soon, the examinee may feel that it is unfair and will complain
about it; he or she will try to explain that if more items would have been presented, the result would be different.

- **The ability level is too far**: In some cases, it could be convenient to stop the test if the ability level is too far from the pass-fail level. This could happen if the examinee responds all items correctly or incorrectly since the beginning of the test. It could indicate that the examinee is an expert in that area, or that he or she does not have any idea about the topics evaluated.

- **Validation**: Some systems could implement an additional validation module to search for patterns within the examinee’s answers. This validation module could recognize simple patterns; for example, when the examinee always answers the same option in a multiple choice test (e.g. always A), or in sequential patterns such as A, B, A, B, A, B. Another situation that could be detected, is when the examinee answers too fast; this could indicate that he or she does not read the question and is guessing.

1.2.3 Item Calibration

Another key aspect of CATs is to have a calibrated item bank. The models described above use one to three parameters, difficulty \( b \), discrimination \( a \), and guessing \( c \) to select items. It is suggested that the item bank has to be as large as possible and the difficulty of the items is widely spread out over the difficulty scale [37].

Classic item calibration in IRT uses likelihood functions such as the Joint Maximum Likelihood (JML), the Conditional Maximum Likelihood (CML) and the Marginal Maximum Likelihood (MML) in order to adjust model parameters. These functions are used to analyse the examinee’s responses and to estimate the item parameters according to them [37, 51, 38]. In order to use this calibration, several previously calibrated items are required, and a significant number of responses from examinees to these items.

Lilley [37] mentions the online item calibration, which consists in using the examinee’s responses of previously calibrated items to estimate parameters of new items along the test. Additionally, this online calibration can be used to refine existing IRT parameters.

However, to generate a new item bank the approaches described above can not be used. In this case, a calibration carried out by experts is performed; they set all item parameters based in their experience with tests (paper and pencil or computer) [37, 38].

1.3 Adaptability in E-Learning

An adaptive learning system could support learners that have different goals, learning styles, knowledge and background. In addition to this, the system assists content navigation, suggesting paths to students (not necessarily linear). Hauger and Kock present a state of the art on adaptability in E-Learning [28], they review some E-Learning systems with and without adaptability; this review includes popular systems such as Blackboard [29] and Moodle [1]. Table [1.1] summarizes this revision and the different features of the E-Learning systems.

Morales *et al.* present a framework for computer adaptive testing based on Service Oriented Architecture [45]. The framework has the following steps: creation of the item
<table>
<thead>
<tr>
<th>Feature</th>
<th>Bedington</th>
<th>Dokee</th>
<th>Moodle</th>
<th>OLAT</th>
<th>OpenUSS</th>
<th>Sakai</th>
<th>AHA</th>
<th>Alf.net</th>
<th>ELM-ART</th>
<th>InterBook</th>
<th>KnowledgeSea</th>
<th>METOD</th>
<th>Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignments-Choice</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignments-Text</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignments-Files</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stored Evaluation</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glossary</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bookmarks</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Annotations</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cooperation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forum</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>News</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-mail</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personal Messages</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Management</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calendar</td>
<td>*</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polls</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nickpage Authoring</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collaboration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chat</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio Conferencing</td>
<td>*</td>
<td>X</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video Conferencing</td>
<td>*</td>
<td>X</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whiteboard</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desktop Sharing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Group Support</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating Content Online</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Importing Content</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stored Learner Model</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personalization</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptive Guiding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link Annotation</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link Hiding</td>
<td>X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptive Pres. Support</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standards</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMS CP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMS QTI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCORM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEEE LOM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1. X → Supported, * → Planned to be supported, ** → not part of the original system, but added as an extension
bank, item packing, metadata addition to items, definition of selection and ordering rules, and, finally, generation of an XML file which describes how to dynamically build the test. Additionally, this framework offers personalized learning routes. In order to use these routes in a course, the professor has to design the course’s main goal and objectives in terms of competencies.

Cisar et al. present a computer adaptive test applied to a C++ exam [13]. They point out some of the advantages and limitations of CATs. The Advantages which are summarized below:

**Advantages**

- Tests are given “on demand”.
- Results are available immediately.
- Answer sheets and test administrators are not needed.
- Computer testing provides a number of configuration options for the questions.
- Usually CATs take less time than regular tests (in a CAT test, fewer items are needed to have acceptable accuracy).
- CATs can provide accurate scores over a wide range of abilities, while traditional tests are usually most accurate for average examinees.

**Limitations**

- CATs cannot be applied for all subjects and skills.
- Hardware could restrict the type of items that can be administered by a computer.
- CATs require careful item calibration.
- Each examinee takes a different set of questions, and may perceive inequities.
- Usually, examinees are not allowed to go back and change answers.

Additionally, Cisar et al. highlights that one of the most important things in the administration of CATs, is how to place the examinee into an ability scale. Their system is developed in Matlab, they used a maximum likelihood function in order to the calculate examinee’s ability. They divided questions in three groups: easy, medium and hard. If the examinees answer a question correctly, the algorithm increase the difficulty of the next one; if he or she answers incorrectly, the algorithm decrease the difficulty of the question.

Lee et al. present a web-program in order to estimate the examinee’s ability, based on IRT-CAT [35]. They included the Rasch model (or 1PL model), and the Two and Three-Parametric Logistics models. The authors highlight that a system requires a examination management system (including an item bank), a system of parametric estimation according to IRT, a system of examinee’s ability estimation, and a system of reevaluation of items after termination of examination. In their work, Lee et al. include item information such as: category, item type, IRT model and item parameters, among others. They use a maximum likelihood function in order to estimate the examinee’s ability for Rasch model, Two or Three Parametric Logistic models.
1.4 E-Learning in Colombia

Some institutions in Colombia have worked in the E-Learning area. The SENa (Servicio Nacional de Aprendizaje, or National Learning Service, in English) uses online learning platforms such as Blackboard\textsuperscript{29} for online courses and as a support for some traditional academic programs\textsuperscript{61}. The Ministerio de Educación Nacional (Ministry of Education), through the Colombia Aprende web, has attempted to establish a website for online courses\textsuperscript{3}. The E-Learning Colombia Union leaded by the Universidad Autónoma de Bucaramanga, UNAB Virtual\textsuperscript{34}, works in a similar way. The Universidad Tecnológica de Pereira, with Mirianda X platform\textsuperscript{15}, is the first MOOC in Colombia. The Universidad Nacional de Colombia has some research projects in its campuses and in the office of virtual national academic services\textsuperscript{16, 67}. The Universidad Abierta y a Distancia has different distance education processes\textsuperscript{66}.

An Intelligent E-Learning Platform is being developed at the Universidad Nacional de Colombia\textsuperscript{24, 23}; it supports the learning process of students with different backgrounds at the university. The computer programming course is being used to proof the concept of the platform; it has three main modules: a Content Management Module, a Learning Management Module and a Testing Module. A course can be defined with a nonlinear structure of content units expressed as a directed graph. Each content unit is defined with a set of short videos, book chapters, presentations and tests.

In the platform content management module a course is defined as a set of content units. These have dependencies in terms of time, organization of the course or content prerequisites. The course is represented in the platform as a directed graph (map) that shows dependencies between content units. This suggests students a way to navigate the course; e.g. given three topics of the course: Languages, Logic and Sets, topics can be ordered as follows: Sets depend on Logic and Logic depends on Languages; in this way, the graph would be Languages $\rightarrow$ Logic $\rightarrow$ Sets (Figure 1.7 shows the Computer Programming Course map). The map idea is to give students freedom to choose the topic of his or her preference. The base component of the map is the “HUB”. This is defined as a point in the space in which students start a course (spaceship in Figure 1.7). Students can go from this point to any content unit, which are represented by a planet. If a content unit is available from another, the planet is surrounded by stars in order to guide students. Content units can be organized in two ways: manual and automatic. In manual organization the course coordinator put every planet in the position he or she wants. Automatic organization creates the map based topic dependencies. Once the map is defined, students can navigate through it and access all content units.

The platform learning management module presents the content of the topics in a “classes” schema, which is based on a series of materials that include videos (no longer than 10 minutes), documents and exercises, created by the teacher. The resources of a content unit are presented in a timeline, which suggests a navigation sequence between them; however, students are able to access resources as they please. Additionally, each class includes a chat-room. The idea of this chat is to simulate the collaborative communication in a traditional classroom. Chat supports sending messages to everyone in the class (public messages), or to a specific user (private messages).
1.5 Data Mining in E-Learning

The data mining process consists in the extraction of unknown knowledge from a large number of incomplete, noisy and ambiguous data [64]. In order to discover knowledge from that data, the following five steps are carried out: problem analysis, data collection, data processing, establishment of the model and assessment of the model [11, 27, 64]. There exists a wealth of techniques and methods to mine data, which in Data Mining are classified in the following categories:

- **Prediction**: Is the inference of a single aspect of the data from a combination of other aspects of it, such as an analysis of the students performance. There are three types of prediction techniques: Classification, regression and density estimation.

- **Clustering**: Is to group objects into classes of similar objects. In educational data, mining clustering has been used to group students according to their behavior.

- **Relationship mining**: Its goal is to discover relationships between variables in a data set with a large number of variables, such as association rule mining, correlation mining, sequential pattern mining, and casual data mining. Within relationship mining, the following subareas are found:
  
  - **Association rule mining** discovers relationships among attributes in data set, producing if-then statements concerning attribute-values.
  - **Correlation mining** finds linear correlations between variables.
  - **Sequential pattern mining** is a more restrictive form of association rule mining in which the accessed item’s order is taken into consideration.
  - **Casual data mining** attempts to find whether one event was the cause of another event.

- **Other techniques**
  
  - **Outlier detection** discovers data points that are significantly different than the rest of the data.
Social network analysis is a field of study attempting to understand and measure relationships between entities in networked information.

Every day more and more data is put into web pages: personal data, social data, education data, etc. This data could be found directly into the web page or indirectly (such as web activity) [27], e.g. content of web pages, intra-page structures, inter-page structure (navigation), usage data and user profiles, among others. In order to understand and get useful information with these data, data mining techniques are used in different processes of the web, e.g. search processes made by search engines could be used for personalization processes, for modifying content according to users needs, for recommendation processes or to detect irregularities [11, 52]. When data mining is used along with web data this is called web mining. Figure 1.8 showed a taxonomy of web mining made by Hanna [27], the taxonomy is divided into three main categories: web content mining, web structure mining and web usage mining. Some specific processes of these categories are highlighted in some subcategories such as: web page content mining, search mining, general access pattern tracking and customized usage tracking, each within its own category.

![Web mining structure](image)

In particular, web usage mining can be used to generate custom education through adding an individual path for each student and by supporting the personalization process in education. Additionally, web content mining and web structure mining can be used to organize and to structure information from the web through discovering association rules, identifying patterns or reorganizing content [11].

The web mining can be used with data generated by E-Learning systems. These have two main users that provide data, students and teachers. Students provide personal information to be stored, in some cases their personal interests, and historical information: their grades, courses, etc. Teachers provide information such as: general information (name, type, company or school), courses and its information, courses content, etc.

The web mining can study some E-Learning problems such as classification problems or clustering problems [11]. The classification problems aim to model the existing relations between multivariable data into classes or labels. Some data mining methods can be used to manage those problems, e.g. fuzzy logic methods, artificial neural networks, evolutionary computation, graphs and trees, association rules, multi-agents systems, etc. In the other hand, the clustering problems in E-Learning consist in find and model groups of data according to a similarity measure. Other data mining methods including prediction techniques, visualization techniques, statistics, semantic web technologies, ontologies, case-
based reasoning and theoretical modern didactic approaches can be used in E-Learning
problems.

From the E-Learning point of view, there are some aspects that can be supported by
the use of data mining techniques. The following five aspects are highlighted by Castro et
al. [11]:

- Applications dealing with the assessment of student’s learning performance.
- Applications that provide course adaptation and learning recommendations based
  on the student’s learning behavior.
- Approaches dealing with the evaluation of learning material and educational web-
  based courses.
- Applications that involve feedback to both teachers and students of E-Learning
courses, based on the students learning behaviors.
- Development determining of atypical students learning behavior.

Castro et al. [11] remark that the assessment of students in E-Learning is the most
commonly studied area where data mining methods are used. Additionally, one of the
topics with least studies is the analysis of atypical student’s learning behavior. They
highlight the following areas as important future trends in E-Learning and data mining:

- E-learning course optimization.
- Student’s E-Learning experience improvement.
- Support tools for E-Learning tutors.
  - Tools used to evaluate students learning performance.
  - Tools that allow performing an evaluation of the learning materials.
  - Tools that provide feedback to the tutors based on the student’s learning be-
    havior.

Some E-Learning systems that use data mining techniques are illustrated by Castro et
al. [11]; Table 1.2 shows these systems.

Two examples of E-Learning applications that use data mining techniques in order to
improve students experiences, are presented by Ribeiro [52] and Zhou [69].

Ribeiro [52] uses a weblog-base framework to analyze the student’s navigational behav-
ior in the evaluation phase of an E-Learning course. The framework uses data mining to
analyze, process, and explore data in order to discover consistent patterns and systematic
relationships between variables, and then validate the results using the detected patterns
to new subsets of data. Ribeiro considers factors such as: the time students take to solve
a quiz (in minutes), the number of hours of study for a quiz and the grade in quizzes to
generate a classification of students. With the results of this classification, students obtain
the course grade (pass or fail).

Zhou [69] presents the application of sequential data mining algorithms in order to
analyze computer logs and create learner’s profiles in terms on their learning methods.
Zhou uses sequential pattern analysis to evaluate student’s activities, and uses that information to customize resource delivery. After that process, Zhou compares detected behavioral patterns with expected educational interfaces and made suggestions to learners with similar characteristics. The detection of these behavior patterns allowed Zhou to personalize activities, to distinguish groups of learners, to identify interaction sequences which indicate problems and patterns that denote success, to recommend learners topics to take, to improve E-Learning systems, and to identify predictors of success by analyzing transfer student records.

In particular, clustering techniques and association rules are regularly used with E-Learning data. Algorithms such as K-means can be used to create clusters using data from an E-Learning system; this cluster can represent students profiles or content units accord-
ing to its features. Association rules can be used to discover hidden relations between transactions over an E-Learning system.

1.5.1 K-Means Algorithm

Given a database $D$ with $n$ objects, K-Means is a partitioning algorithm that classifies these $n$ objects into $k$ partitions; where $k \leq n$, each partition represents a cluster. K-Means classifies the objects into $k$ clusters where inner cluster similarity is low and intra cluster similarity is high, i.e. all object inside a cluster are similar to each other, but are different in comparison to objects from other clusters. Cluster similarity is measured based on the mean value object in the clusters (usually using the euclidean distance) [49].

K-Means starts with a random selection of $k$ objects, where each one represent a cluster mean or centroid. All the remaining objects are assigned to the most similar cluster, based on the measure between the object and the cluster mean. When an object is added to a cluster a new cluster, mean is calculated. This process iterates until a criterion function converges; usually the Square Error is used. Equation (1.16) shows the Square Error metric, where an object located in space is represented by $x$, and $m_i$ is the mean of the cluster $C_i$ ($x$ and $C_i$ are multidimensional). The Square Error tries to make the final $k$ clusters as compact and as separate as possible[49].

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - m_i|^2 \quad (1.16)$$

Besides the Square Error there are other metrics used to measure how compact is a clustering process. Two of those metrics are the Silhouette Coefficient and the Davies Bouldin Index.

1.5.1.1 Silhouette Coefficient

The Silhouette Coefficient uses the average proximity of the cluster in order to establish a metric. After a clustering process, for example k-means, a group of clusters is obtained. Given two clusters, $A$ and $B, A \neq B$, and $i$ being an element within $A$, the average dissimilarity of $i$ to all elements within $A$ is calculated, it is referred as $a(i)$. The dissimilarity of $i$ to all elements of $B$, is now referred as $b(i)$. The Silhouette Coefficient is calculated based on $a(i)$ and $b(i)$ by using the equations (1.17) or (1.18).

$$S(i) = \begin{cases} 
1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\
0 & \text{if } a(i) = b(i) \\
\frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) 
\end{cases} \quad (1.17)$$

$$S(i) = \frac{b(i) - a(i)}{\max \{a(i), b(i)\}} \quad (1.18)$$
1.5.1.2 Davies Bouldin Index

The Davies Bouldin Index is another metric used to calculate the similarity between clusters [14]. $S_i$ is the dispersion measure of the cluster and is defined in Equation (1.19); usually $q = 2$ in order to use the euclidean distance.

$$S_i = q \sqrt{\frac{1}{T_i} \sum_{j=1}^{T_i} |X_j - A_i|^q}$$  \hspace{1cm} (1.19)

In Equation (1.19), $A_i$ is the centroid of cluster $C_i$ and $T_i$ is size of cluster $i$. $M_{ij}$ is the separation measure between cluster $C_i$ and cluster $C_j$, which is defined in Equation (1.20) where $a_{ki}$ is the $k^{th}$ element of $A_i$.

$$M_{ij} = \|A_i - A_j\|_p = \sqrt[p]{\sum_{k=1}^{n} |a_{ki} + a_{kj}|^p}$$  \hspace{1cm} (1.20)

$R_{ij}$ (Equation (1.21)) is the measure of how good is the cluster. This measure uses $M_{ij}$ as a separation of the $i^{th}$ element and the $j^{th}$ cluster:

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}$$  \hspace{1cm} (1.21)

$\bar{R}$ is the system large average of the similarity means within the most significant cluster, and it is defined in Equation (1.22), where $R_i = max(R_{ij}, i \neq j)$:

$$\bar{R} = \frac{1}{N} \sum_{i=1}^{N} R_i$$  \hspace{1cm} (1.22)

1.5.2 Association Rules

Association rules are a Data Mining method that is used to discover significant relations between items or variables in databases or warehouses [36]. The goal of association rules is to extract strong relations of the form $X \Rightarrow Y$. This relations mean: items which satisfy condition $X$ are most likely to satisfy condition $Y$. Suppose there is a set of $n$ items $I = \{i_1, i_2, ..., i_n\}$, and $T$ is a set of transactions, where each individual transaction $t$ has a set of items $t \subseteq I$, a transaction $t$ contains a subset of items $A$, in which $A \subset t$. Hence, an association rule has the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$ and $A \cap B = \emptyset$.

Association rules have two main properties: support and confidence. Support $s$ of a rule $A \Rightarrow B$ is the $s$ percent of transactions in $T$ that contains the rule. Confidence is a certainty measure for association rules; if $A$ and $B$ are sets of items in any transaction, a $c$ percentage of transactions that contains $A$ and also contains $B$ in the association $A \Rightarrow B$ represents that confidence [36] [49].
CHAPTER 2

Computer Adaptive Tests Process in an E-Learning Platform

2.1 Introduction

Tests are constructed to measure the knowledge level of examinees according to some test-givers requirements, this level could be measured individually or as a group [38]. Establishing if an examinee has the required knowledge to pass a specific test is not an easy task. In order to evaluate examinees, test-givers (professors in charge of creating tests) have to consider some factors while they build tests [38]. For example, if the test is too easy, some of the examinees could consider it a waste of time. This would take them to underestimate questions and make careless mistakes, or may fall into “tricky questions” without noticing. On the other hand, if the questions are too hard, the examinee can feel discouraged and in some cases panic or despair, ceasing his or her attempts to seriously answer and just try to guess [38].

Another fundamental factor to be consider in the tests creation, is the length of the test (number of questions to administer). This is directly related with its duration. Professors have to make an estimation of how much time it takes to an examinee to answer each question and the time to complete the test. If it is too long, the examinee may lose concentration and try to guess in the last questions since he or she is tired. On the other hand, if the test is too short, the questions may not reflect all topics that should be evaluated, and some of the examinees may consider it unfair.

So, how many questions should a test have? How easy or hard should they be? Computer Adaptive Tests (CAT) propose a solution to these questions and to estimate the examinee’s ability based on his or her progress in the test. CATs, as its name implies, is a software application where test content (items) adapts to each individual examinee [37]. A CAT works based on a group of questions called item bank. Each item in the bank has a difficulty level (another parameter could be included); for instance, one question of a computer programming test is the following:

*Given the following C++ program:*

```
1  #include <iostream>
2  using namespace std;
```
# CHAPTER 2. COMPUTER ADAPTIVE TESTS PROCESS IN AN E-LEARNING PLATFORM

```c
int main(){
    int a, b, c, m, n, p;
    a=2;
    b=3;
    c=5;
    m=b/c*a+b-c%b/a;
    n=c/4*6*m/a-b+c;
    p=m*b/n*b/a+5-(-c*n);
    cout << m << " " << n << " " << p;
    system("pause");
    return 0;
}
```

The values in the screen are:

A. -1 -1 0
B. 0 2 15
C. -1 -1 4
D. -1 -1 -4

The question above has a difficulty $X$ between $[-3, 3]$, this scale is based on Lilley’s work [37]. If an examinee answers the question correctly, his or her ability level is higher than the item’s difficulty $X$. In this case a harder item is presented. In the opposite case, when an examinee answers incorrectly, his or her ability level is lower than the item’s difficulty $X$. In this way, an easier question would be presented. This process is repeated until the test ends. Figure 2.1 shows the general process of a CAT, the red line indicates the ability level of the examinee and the blue line the difficulty of questions which after a series of questions converges to the examinee’s ability.

![Figure 2.1. Computer Adaptive Test process](image)

In this chapter, the creation process of the Computer Adaptive Tests Model is presented, including, the creating of an item bank. In the first place, a Basic Model that
uses Item Response Theory models is proposed. This model is tested with some trials, which provides feedback from students and professors. This feedback is used to improve the model in a second version. This version is called Question History model, is tested and is once again improved by using new feedback from students and professors. This model present some issues that are tackled in the final version of the model. This is called Topic History model. This final model is currently used in the Intelligent E-Learning Platform of the Universidad Nacional de Colombia. In order to make the trials, two modules are created in the E-Learning Platform: the test management module and the test taking module. The test management module is used to create items, group them and assign them to a test; additionally, each test can be configured and scheduled. The test taking module is used by students to take tests, it is in charge of the three main processes in the model: item selection process, ability estimation and grading.

2.2 Item Creation

The E-Learning Platform defines four user roles: administrator, coordinator, professor and student. Two of these roles interact with the test management module (coordinator and professor). The test management module allows the creation of tests from scratch. The interface of this module divides the creation of a test in three steps: questions, bags, and tests. Each one of these steps has four basic operations: create, search, edit and delete. In order to build a test, the first step is to create questions, this step can be done by professors and coordinators.

In order to create questions, professors (or coordinators) have to fill the following basic information: question name, description, topic and type. The platform includes four types of questions: matching questions, multiple choice with unique answer, multiple choice with multiple answer and true-false questions. When a professor chooses one of these types, the interface shows the fields that he or she has to fill in order to complete the question, this includes the content of every option (when it applies) and the selection of the correct answer for the question. Additionally, in order to use CATs, professors have to fill the three parameters for the IRT models (which are difficulty, discrimination and guessing). The questions interface is illustrated in Figure 2.2.

After the question creation process, the course coordinator creates bags to group questions (only the coordinator can create bags and tests). These bags associate questions according to the coordinator criteria. The coordinator has two options for bag creation: automatic or manual. To make bags automatically, the coordinator uses four filters (question type, question difficulty, topic or name) in order to define him or her criterion, and questions that match that criterion are grouped in a bag, each bag has a name and description. In the manual function the coordinator uses the filters mentioned before as a initial search, and then he or she picks which specific questions are added to the bag. Figure 2.3 shows the bags interface using the manual function.

After the bags are done the coordinator is ready to build the test. In order to create one, the coordinator has to fill some basic information (name and description) and assign one or more bags to the test. Additionally, the coordinator has to select how the text will be presented, with a random selection of questions or adaptive selection. As its name implies, random selection chooses a question, randomly, from the test bags every time a

\footnote{Available at \url{http://www.alife.unal.edu.co/platform-svn/trunk/}}
student answers one. When random the selection is used, the test ends when all questions in the bags are took. The adaptive selection uses the CAT model proposed in this Thesis in order to select questions. When adaptive selection is selected, the coordinator has to specify which Item Response Theory model is used (1PL, 2PL or 3PL). Additionally, he or she must set a stopping rule for the test from the following options: fixed length, ability estimation (variable length) or combined. When fixed length is selected, the test ends when
a student reaches a maximum number of questions that is defined by the coordinator. If the coordinator chooses the ability estimation, the test ends when a minimum standard error is reached. The standard error is calculated according to the IRT model previously selected, 1PL uses Equation (1.4), 2PL uses Equation (1.9) and 3PL uses Equation (1.14) to calculate it. In addition to this, the coordinator has to set a minimum number of questions in order to be taken before the test ends. If the coordinator chooses the combined option, he or she has to set the parameters of both previous rules, i.e. maximum number of questions, minimum standard error and minimum question to be presented. The test ends when one of these conditions is reached. Figure 2.4 shows part of the test interface for the creation of an adaptive test.

In order to create a new item bank from scratch in the computer programming course, the course coordinator does a “call for items” to experts (professors of the course), and professors upload questions specifying difficulty, discrimination and guessing. To appropriately set the IRT parameters, the models are explained to the professors beforehand. Each one make 10 questions per group taught (they are in charge of one or two groups). Hence, each exam has around 200 questions in its item bank.

Each IRT parameter has two scales, one is used by professors in the questions interface, and another is used in the IRT models. Table 2.1 shows the two scales for each one of the three IRT parameters. Professors use the following scales: difficulty from 1 to 7 (medium difficulty 4), discrimination from 1 to 5 (neutral discrimination 3), guessing from 0 to 1. Each professor sets difficulty and discrimination parameters according to his or her experience. Guessing parameter is set based on the question type. For instance, a multiple choice question with one answer that has four possible options (A, B, C, D), has a guessing probability of 0.25. However, the 0.25 value depends on how balanced are the options. For instance, if one option makes no sense in the context of the question, a student with no knowledge will easily discard this option; for this reason, the question guessing probability is around 0.33.
25

<table>
<thead>
<tr>
<th>IRT Parameter</th>
<th>Interface Scale</th>
<th>Model Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difficulty (b)</td>
<td>1 to 7</td>
<td>-3 to 3</td>
</tr>
<tr>
<td>Discrimination (a)</td>
<td>1 to 5</td>
<td>-2 to 2</td>
</tr>
<tr>
<td>Guessing (c)</td>
<td>0 to 1</td>
<td>0 to 1</td>
</tr>
</tbody>
</table>

Table 2.1. Scales of IRT parameter in the questions interfaces and internally used in IRT models

2.3 Basic Model

The Basic Model has three main stages: initialization, selection and grade. The initialization stage, is in charge of loading the test configuration, which includes, an IRT model (1PL, 2PL or 3PL), a stopping rule and an item bank.

The selection stage is the core of the model; this stage is done while students take a test. Every time a student answers a question, his or her ability is calculated using the Maximum Likelihood Function of the IRT model. This ability value is used to select the next question and administer it to the student. This process continues until the test stopping rule is reached. The selection algorithm is based on Lilley’s work [37]. The ability value is used along with the last answer of the student to search for the next question. This process is carried out as follows: if the last answer is correct, the algorithm looks for the next question with a higher difficulty nearest to the ability value calculated. If the last answer is incorrect, the algorithm looks for the next question with a lower difficulty nearest to the ability value calculated. If the last answer is incorrect, the algorithm looks for the next question with a lower difficulty nearest to the ability value calculated. If it finds only one question that matches this requirement, it is presented to the student. If there is more than one question, the algorithm obtains a set of questions with the same difficulty level that are grouped in an array called arrayB. After this, the algorithm uses the discrimination parameter to search for the question with its higher value within arrayB. If only one question matches that criterion, it is presented to the student. In the other case, there are multiple questions that match that criterion, they are grouped in an array called arrayA, then the algorithm looks for the question with the lower guessing value within arrayA elements and presents it to the student. If it obtains more than one question, the questions are put into an array called arrayC. Finally, the algorithm randomly selects a question within arrayC and presents it to the student. This process describes the selection for the 3PL model. If the course coordinator chooses the 2PL model or the 1PL model, the random selection is made after the last parameter of each model is used; i.e., in 2PL model, the random selection is made using arrayA.

The grade stage is made after the test has reached its stopping condition. In order to obtain the final grade of a student, the grading algorithm uses the final ability value of the student and converts it from its scale (-3 to 3) to the grade scale (0 to 5). If the final ability value is out of the valid range (-3 to 3), the algorithm uses another method to calculate the grade of the student. In this case, the algorithm calculates an extra question (the one that will follow to the last question), uses its difficulty along with the difficulty of the last question that the student took, calculates the average these two values and uses it in order to calculate the student grade. This is carried out by converting the average of the difficulties from its internal scale (-3 to 3) to the grade scale (0 to 5). The general structure of the Basic Model is shown in Figure 2.5.
Some trials are made using the Basic Model in May of 2013 (first test of the computer programming course); for this trials the test has an item bank with 190 questions created by 12 professors. To create that item bank, groups of three professors are made; within each group, professors have to propose and review questions content and their parameters. The 3PL model is used in this and will be used in future trials.

In the trials it is observed, that students are surprised by the new evaluation methodology. They do not understand how their grade is calculated, since they are used to calculate it based on correct answers over total questions. To clarify the situation, professors explain to them how the new evaluation methodology works, and how it calculates grades.

This trial reveals some issues with the test taking module; some of them are technical problems and other are problems of the model. Technical issues include network problems and some minor inconveniences in the test taking interface; the test taking module is modified in order to deal with these technical issues. Additionally, results and complaints of students are reviewed. They revealed three issues with the Basic model:

- If from the beginning of the test a student answers four consecutive questions incorrectly (or correctly), the test is over and the student’s grade is 0 (or 5 respectively).
- In some cases the student ability got stuck after the second question.
- In some cases students answered a question correctly and the ability got lower.

These three issues are analyzed and its causes are discovered. The first issue is caused by the difficulty parameter. Professors set this parameter with integer values (e.g. 1, 2, ..., 7); this means that every time a student answers a question, his or her next question has a difficulty level increase or decrease by one unit. Given that the test starts with a question with medium difficulty, when a student fails three consecutive questions, he or she reaches the lower difficulty level; with his or her fourth incorrect answer, the difficulty level is out of the scale and the test is finished. The same situation happens when there are four consecutive correct answers: the difficulty level is out of the scale in its upper bound. Particularly, only students with a 0 grade complained about this situation.
The second and third issues are related to the discrimination parameter. A revision and analysis of the problematic question is done, and it revealed that some professors created questions with a low difficulty value and a high discrimination value. If a student fails a question with these parameters, the maximum likelihood function “gets stuck”, and the future answers of the student do not matter. The maximum value of the maximum likelihood function has the same $\theta$ (ability) value.

Figure 2.6 and Table 2.2 show the general performance of this trial: 69% of students failed the test, 30% passed it with a grade high than 3, and 1% passed it with a grade equal to 3. Additionally, 60 students obtained a final grade of 0, and 72 students had a grade of 5.

The platform administration module provides a tool that shows in detail the progress of students in a particular test. Professors use this tool in order to show to students their test information, including their grade, the number of questions taken and a graph that shows their progress in the test. Figures 2.8, 2.9, 2.7, 2.10 and 2.11 show some examples of students in this trial. Red points represent the answer of a question, 1 if it is correct and 0 if it is incorrect. The green line represents the ability of the student across the test and blue dots represent the difficulty of a question.

Figure 2.7 presents an example of the regular development of the test. The student starts with an item near medium difficulty, answers it correctly, and so, the second one is more difficult; he then answers incorrectly, and so, the third one is easier, and the process continues that way until the test ends. This is an example of an average student that answers 10 questions and has a grade of 3.33.

Figure 2.8 presents an example of the regular progression of the test where a student obtains a grade of 3.9, it has a little irregularity between questions 6 and 7 when the ability drops suddenly and difficulty of the questions is the same.

Figure 2.9 shows a regular progression of the test where a student obtains a grade of 5. The process of the student is very clear: he correctly answers the firsts three questions, and fails in the fourth one; after that he answers correctly and incorrectly some questions, and finally answers one correctly in the upper bound of the scale. Hence, the test ends.
Figure 2.7. Progress of a student in the first test (I)

Figure 2.8. Progress of a student in the first test (II)

Figure 2.10 shows how the test ends after four consecutive correct answers. Students that answer four questions correctly (starting from the first one), have a grade of 5; they finished the test on the average of a half hour. A similar situation happens when students answer four questions incorrectly. Students that answer incorrectly the first four questions obtain a grade of 0. Averagely, they answers the test in 20 minutes.

Figure 2.11 shows a special case in the progression of a test. This and some other cases are analyzed revealing that the main problem is a series of questions with a low difficulty and a high discrimination. This particular case modifies the regular behaviour of the
likelihood function, which creates irregularities in the ability estimation. For instance, in this case, ability reaches values below 1, something that should not happen.

2.4 Question History Model

In order to deal with the set of issues found in the Basic Model, some improvements that generate a second model called Question History Model are proposed. This model takes Basic Model structure and adds to it a history of questions. This history is used to decide if the difficulty of the next question will change or not (if it uses the adaptive process or
not). When a test starts, the history record initiates empty; when a student answers a question, the system stores the score of the question (1 if its correct or 0 if its incorrect) and presents a question of the same difficulty of the previous one. A search of questions of the same difficulty level is carried out in the item bank within a range close to the difficulty of last one answered. The system presents questions of the same difficulty until the history has 2 out of 3 questions correct or incorrect. When the history has the 2 out of the 3 questions required, the adaptive selection process of the Basic Model is used, and the history record empties. Figure 2.12 shows the structure of the Question History Model; the blue boxes show the components of the Basic Model and the red boxes show the improvements of this model.

In order to tackle previous issues with the discrimination parameter, the irregularities and its causes are explained to the professors. The effect of low difficulty questions with a high discrimination is highlighted. Particularly, this configuration is illogical since an easy question does not establish a significant difference between the knowledge of two students, therefore, this configuration should not be valid. To guarantee that this configuration is not set in the new trial or in future ones, the question creation interface is modified including a validation over the discrimination value.

This model is tested with the second exam of the computer programming course in the first semester of 2013 (same academic period of the previous trial). For this trial, a new item bank is created. The improvements integrate in this model can be seen in the progression of students. Figure 2.13 shows an example of the progression of a student in the trial of this model. The student starts the test with a medium level question and the history record starts empty ($qh = [0]$); then he answers the first question correctly, hence, this value is stored in the historical $qh = [1]$; after that, the system presents a question of the same difficulty level (medium); the second question is answered incorrectly, then the historical stores the answer, $qh = [1, 0]$; and then, the system selects a question with the same difficulty level again; the third question is answered correctly, then $qh = [1, 0, 0]$; at
this point, the system uses the adaptive selection and empties the history record $qh = \emptyset$. This process continues until the test reaches its stopping rule.

<table>
<thead>
<tr>
<th>Grade:</th>
<th>3.1667</th>
<th>Number of question:</th>
<th>10</th>
</tr>
</thead>
</table>

Figure 2.12. Question History Model

Figure 2.13. Progress of a student in the second test

Figure 2.14 and Table 2.3 show the general performance of the trial of the Question History Model. A 31% of the students passed the test with a grade higher that 3; 6% passed it with a grade equal to 3 and 63% failed the test. Additionally, almost 100 students did not take the test because they canceled the course. In general, statistics are similar to the first trial (trial of the Basic Model). However, the number of students that got a grade equal to 0 and 5 decreased.
Individual and general results of the test are analyzed, and it can be observed that the question history record helps to balance the test. Additionally, in the development of the trial, it can be observed that changes introduced in this model make students feel more comfortable with the evaluation process. Moreover, students now have previous experience with the evaluation methodology and understand it better; in this trial, a few students complained about their grade, but professors explained to them their exact progress and their doubts are solved.

After the test, students pointed out an issue that this model does not take into account. They said that they studied all subjects for the test, but only answered questions of a few of them, in some cases only one. This issue is discussed with the professors of the course and a solution to it is presented in the next version of the model.

![Figure 2.14. General performance of second test (2013-I)](image)

<table>
<thead>
<tr>
<th>Students</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>421</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>38</td>
<td>= 3</td>
</tr>
<tr>
<td>204</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>23</td>
<td>= 0</td>
</tr>
<tr>
<td>5</td>
<td>= 5</td>
</tr>
</tbody>
</table>

**Figure 2.14. General performance of second test (2013-I)**

**Table 2.3. General grades of second test (2013-I)**

### 2.5 Topic History Model

The issue that is discovered in the previous model is considered and some changes are made in order to deal with it. The idea of the historical array that is introduced in the previous model, is used to create a topic history record. This is used to search for and show questions of all topics within a test. Figure [2.15](image) shows the improvements introduced in this model in the green boxes. These changes modify in the selection stage, when a student answers a question, the topic of that question is stored in a historical called \( th \). When the system looks a question, it searches for one of a topic that is not in the array \( th \). The array \( th \) is used in both selection methods, adaptive and same difficulty. If the algorithm does not find a question with different topic of the \( th \) content, it removes the first item in \( th \) and searches again. When a student takes questions of all topics, \( th \) is emptied.

In 2013-I period due to external reasons (social protests) the term was not carried on in a regular way. This interfered with the third test of the computer programming course in that academic period. Therefore, this model is tested in the next period (2013-II). The 2013-II period had three tests that are used as trials for this version of the model. With a new academic period, new students take the course and test this model, even though, a few students repeat the course.
Results from the first test of 2013-II period are shown in Figure 2.16 and in Table 2.4. They show that a 58% of the students passed the test with a grade higher than 3; 5% passed it with a grade equal to 3 and 37% failed the test. These results show an increase on the percentage of students that passed the test, compared to the previous trials (first and second test of 2013-I). The increase of students that passed the test is given by three circumstances:

- The model is progressively upgraded to cover the requirements and suggestions of students and professors.
- The creation process of the item bank is easier every time; professors have learned how to establish the parameters more accurately every time.
- Students are getting used to the evaluation method. They speak with other students (from previous semesters) about how the exam will be. Additionally, some students that cancel the course during a semester, take it again the next academic period and share their experiences with other students.

Some students that took the first test did not take the second one, and some that took the second did not take the third in the 2013-II period (similar to the 2013-I). Hence, the number of students in the trials decreased in almost 200 from the first to the third trial.

Results from the second test in the 2013-II period are shown in Figure 2.17 and in Table 2.5. In this trial, 51% of students passed the test with a grade higher than 3; 6%
Results from the second and third test of 2013-II, show a similar behavior to each other and to the first test. The general results of students are similar from the three test in 2013-II, which indicates a consistent behavior of the model. The general grades of students, shows that between 57% and 63% of students pass tests in 2013-II. These results show a general improvement of grades of students compared to the 2013-I period. Additionally, the number of students with a final grade of 0 is lower than tests in 2013-I. Hence, the addition of the topic history improves the model.
2.6 Summary

This chapter presented the designing and building of a Computer Adaptive Test Model. It is integrated into the Intelligent E-Learning Platform of the Universidad Nacional de Colombia and the Computer Programming course is to proof the concept of the platform. It allows the easy use of the model in future versions of the course or in other courses. The proposed CAT model includes 1PL, 2PL and 3PL models from Item Response Theory in its selection process.

The development of the model is presented including its three versions: Basic Model, Question History Model and Topic History Model. The different versions of the model are created by using feedback from students and professors in different trials. The final version of the model includes two historical arrays that tackle some key issues found in the trials. The first array (question historical record) makes the difficulty change slower, this is done by choosing between two selection methods: the selection of an item with the same difficulty of the previous one, or the adaptive selection method. Additionally, the question history prevents that the tests ends when four consecutive questions are answered correct or incorrect (starting from the first one). The second array (topic historical record) guarantees that all topics in a test are evaluated. Results show that these changes improve the general results of students in tests.

The Computer Programming course is an example of how a course evaluation methodology can be changed from a standard evaluation methodology to an adaptive one. Additionally, it is important to point out that students and professors take some time to adjust to the new evaluation methodology.

The creation of item banks is done by using the experts approach. This approach allow courses that do not have a item bank, create one in a short period of time, for instance two week are required to create the item bank for the Computer Programming course.

Part of this chapter will be published in an article entitled “Evolution of Teaching and Evaluation Methodologies: The Experience in the Computer Programming Course at Universidad Nacional de Colombia” in the Revista de Ingeniería e Investigación de la Universidad Nacional de Colombia (Journal of Engineering Research of the National Uni-

Figure 2.18. General performance of third test (2013-II)

Table 2.6. General grades of third test (2013-II)

<table>
<thead>
<tr>
<th>Students</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>163</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>24</td>
<td>= 3</td>
</tr>
<tr>
<td>230</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>4</td>
<td>= 0</td>
</tr>
<tr>
<td>52</td>
<td>= 5</td>
</tr>
</tbody>
</table>
Moreover, the platform development is published in two articles, the first one entitled “A Didactic E-Learning Platform with Open Content Navigation and Adaptive Exercises” in the proceedings of the International Conference on Education and E-Learning Innovations (ICEELI) [24], and the second one entitled “Plataforma de Aprendizaje y Cursos Masivos Abiertos en Línea de la Universidad Nacional de Colombia” (in english, “Learning Platform and Massive Open Online Courses at Universidad Nacional de Colombia”) in the proceedings of Virtual Educa 2013 [23].
CHAPTER 3

Computer Adaptive Tests Process in an E-Learning Platform

3.1 Introduction

Tests are constructed to measure the knowledge level of examinees according to some test-givers requirements, this level could be measured individually or as a group [38]. Establishing if an examinee has the required knowledge to pass a specific test is not an easy task. In order to evaluate examinees, test-givers (professors in charge of creating tests) have to consider some factors while they build tests [38]. For example, if the test is too easy, some of the examinees could consider it a waste of time. This would take them to underestimate questions and make careless mistakes, or may fall into “tricky questions” without noticing. On the other hand, if the questions are too hard, the examinee can feel discouraged and in some cases panic or despair, ceasing his or her attempts to seriously answer and just try to guess [38].

Another fundamental factor to be consider in the tests creation, is the length of the test (number of questions to administer). This is directly related with its duration. Professors have to make an estimation of how much time it takes to an examinee to answer each question and the time to complete the test. If it is too long, the examinee may lose concentration and try to guess in the last questions since he or she is tired. On the other hand, if the test is too short, the questions may not reflect all topics that should be evaluated, and some of the examinees may consider it unfair.

So, how many questions should a test have? How easy or hard should they be? Computer Adaptive Tests (CAT) propose a solution to these questions and to estimate the examinee’s ability based on his or her progress in the test. CATs, as its name implies, is a software application where test content (items) adapts to each individual examinee [37]. A CAT works based on a group of questions called item bank. Each item in the bank has a difficulty level (another parameter could be included); for instance, one question of a computer programming test is the following:

```
Given the following C++ program:

1  #include <iostream>
2  using namespace std;
```
int main(){
    int a, b, c, m, n, p;
    a = 2;
    b = 3;
    c = 5;
    m = b / c + a * b - c % b / a;
    n = c / 4 * 6 * m / a - b + c;
    p = m * b / n * b / a + 5 - (-c * n);
    cout << m << " " << n << " " << p;
    system("pause");
    return 0;
}

The values in the screen are:
A. -1 -1 0
B. 0 2 15
C. -1 -1 4
D. -1 -1 -4

The question above has a difficulty $X$ between $[-3, 3]$, this scale is based on Lilley’s work \cite{37}. If an examinee answers the question correctly, his or her ability level is higher than the item’s difficulty $X$. In this case a harder item is presented. In the opposite case, when an examinee answers incorrectly, his or her ability level is lower than the item’s difficulty $X$. In this way, an easier question would be presented. This process is repeated until the test ends. Figure 3.1 shows the general process of a CAT, the red line indicates the ability level of the examinee and the blue line the difficulty of questions which after a series of questions converges to the examinee’s ability.

In this chapter, the creation process of the Computer Adaptive Tests Model is presented, including, the creation of an item bank. In the first place, a Basic Model that
uses Item Response Theory models is proposed. This model is tested with some trials, which provides feedback from students and professors. This feedback is used to improve the model in a second version. This version is called Question History model, is tested and is once again improved by using new feedback from students and professors. This model present some issues that are tackled in the final version of the model. This is called Topic History model. This final model is currently used in the Intelligent E-Learning Platform of the Universidad Nacional de Colombia. In order to make the trials, two modules are created in the E-Learning Platform: the test management module and the test taking module. The test management module is used to create items, group them and assign them to a test; additionally, each test can be configured and scheduled. The test taking module is used by students to take tests, it is in charge of the three main processes in the model: item selection process, ability estimation and grading.

3.2 Item Creation

The E-Learning Platform defines four user roles: administrator, coordinator, professor and student. Two of these roles interact with the test management module (coordinator and professor). The test management module allows the creation of tests from scratch. The interface of this module divides the creation of a test in three steps: questions, bags, and tests. Each one of these steps has four basic operations: create, search, edit and delete. In order to build a test, the first step is to create questions, this step can be done by professors and coordinators.

In order to create questions, professors (or coordinators) have to fill the following basic information: question name, description, topic and type. The platform includes four types of questions: matching questions, multiple choice with unique answer, multiple choice with multiple answer and true-false questions. When a professor chooses one of these types, the interface shows the fields that he or she has to fill in order to complete the question, this includes the content of every option (when it applies) and the selection of the correct answer for the question. Additionally, in order to use CATs, professors have to fill the three parameters for the IRT models (which are difficulty, discrimination and guessing). The questions interface is illustrated in Figure 2.2.

After the question creation process, the course coordinator creates bags to group questions (only the coordinator can create bags and tests). These bags associate questions according to the coordinator criteria. The coordinator has two options for bag creation: automatic or manual. To make bags automatically, the coordinator uses four filters (question type, question difficulty, topic or name) in order to define him or her criterion, and questions that match that criterion are grouped in a bag, each bag has a name and description. In the manual function the coordinator uses the filters mentioned before as a initial search, and then he or she picks which specific questions are added to the bag. Figure 2.3 shows the bags interface using the manual function.

After the bags are done the coordinator is ready to build the test. In order to create one, the coordinator has to fill some basic information (name and description) and assign one or more bags to the test. Additionally, the coordinator has to select how the text will be presented, with a random selection of questions or adaptive selection. As its name implies, random selection chooses a question, randomly, from the test bags every time a

1 Available at http://www.alife.unal.edu.co/platform-svn/trunk/
student answers one. When random the selection is used, the test ends when all questions in the bags are took. The adaptive selection uses the CAT model proposed in this Thesis in order to select questions. When adaptive selection is selected, the coordinator has to specify which Item Response Theory model is used (1PL, 2PL or 3PL). Additionally, he or she must set a stopping rule for the test from the following options: fixed length, ability estimation (variable length) or combined. When fixed length is selected, the test ends when
a student reaches a maximum number of questions that is defined by the coordinator. If the coordinator chooses the ability estimation, the test ends when a minimum standard error is reached. The standard error is calculated according to the IRT model previously selected, 1PL uses Equation (1.4), 2PL uses Equation (1.9) and 3PL uses Equation (1.14) to calculate it. In addition to this, the coordinator has to set a minimum number of questions in order to be taken before the test ends. If the coordinator chooses the combined option, he or she has to set the parameters of both previous rules, i.e. maximum number of questions, minimum standard error and minimum question to be presented. The test ends when one of these conditions is reached. Figure 2.4 shows part of the test interface for the creation of an adaptive test.

In order to create a new item bank from scratch in the computer programming course, the course coordinator does a “call for items” to experts (professors of the course), and professors upload questions specifying difficulty, discrimination and guessing. To appropriately set the IRT parameters, the models are explained to the professors beforehand. Each one makes 10 questions per group taught (they are in charge of one or two groups). Hence, each exam has around 200 questions in its item bank.

Each IRT parameter has two scales, one is used by professors in the questions interface, and another is used in the IRT models. Table 2.1 shows the two scales for each one of the three IRT parameters. Professors use the following scales: difficulty from 1 to 7 (medium difficulty 4), discrimination from 1 to 5 (neutral discrimination 3), guessing from 0 to 1. Each professor sets difficulty and discrimination parameters according to his or her experience. Guessing parameter is set based on the question type. For instance, a multiple choice question with one answer that has four possible options (A, B, C, D), has a guessing probability of 0.25. However, the 0.25 value depends on how balanced are the options. For instance, if one option makes no sense in the context of the question, a student with no knowledge will easily discard this option; for this reason, the question guessing probability is around 0.33.
### Table 3.1. Scales of IRT parameter in the questions interfaces and internally used in IRT models

<table>
<thead>
<tr>
<th>IRT Parameter</th>
<th>Interface Scale</th>
<th>Model Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difficulty ((b))</td>
<td>1 to 7</td>
<td>-3 to 3</td>
</tr>
<tr>
<td>Discrimination ((a))</td>
<td>1 to 5</td>
<td>-2 to 2</td>
</tr>
<tr>
<td>Guessing ((c))</td>
<td>0 to 1</td>
<td>0 to 1</td>
</tr>
</tbody>
</table>

#### 3.3 Basic Model

The Basic Model has three main stages: initialization, selection and grade. The initialization stage is in charge of loading the test configuration, which includes, an IRT model (1PL, 2PL or 3PL), a stopping rule and an item bank.

The selection stage is the core of the model; this stage is done while students take a test. Every time a student answers a question, his or her ability is calculated using the Maximum Likelihood Function of the IRT model. This ability value is used to select the next question and administer it to the student. This process continues until the test stopping rule is reached. The selection algorithm is based on Lilley’s work [37]. The ability value is used along with the last answer of the student to search for the next question. This process is carried out as follows: if the last answer is correct, the algorithm looks for the next question with a higher difficulty nearest to the ability value calculated. If the last answer is incorrect, the algorithm looks for the next question with a lower difficulty nearest to the ability value calculated. If it finds only one question that matches this requirement, it is presented to the student. If there is more than one question, the algorithm obtains a set of questions with the same difficulty level that are grouped in an array called `arrayB`. After this, the algorithm uses the discrimination parameter to search for the question with its higher value within `arrayB`. If only one question matches that criterion, it is presented to the student. In the other case, there are multiple questions that match that criterion, they are grouped in an array called `arrayA`, then the algorithm looks for the question with the lower guessing value within `arrayA` elements and presents it to the student. If it obtains more than one question, the questions are put into an array called `arrayC`. Finally, the algorithm randomly selects a question within `arrayC` and presents it to the student. This process describes the selection for the 3PL model. If the course coordinator chooses the 2PL model or the 1PL model, the random selection is made after the last parameter of each model is used; i.e., in 2PL model, the random selection is made using `arrayA`.

The grade stage is made after the test has reached its stopping condition. In order to obtain the final grade of a student, the grading algorithm uses the final ability value of the student and converts it from its scale (-3 to 3) to the grade scale (0 to 5). If the final ability value is out of the valid range (-3 to 3), the algorithm uses another method to calculate the grade of the student. In this case, the algorithm calculates an extra question (the one that will follow to the last question), uses its difficulty along with the difficulty of the last question that the student took, calculates the average these two values and uses it in order to calculate the student grade. This is carried out by converting the average of the difficulties from its internal scale (-3 to 3) to the grade scale (0 to 5). The general structure of the Basic Model is shown in Figure 2.5.
Some trials are made using the Basic Model in May of 2013 (first test of the computer programming course); for this trials the test has an item bank with 190 questions created by 12 professors. To create that item bank, groups of three professors are made; within each group, professors have to propose and review questions content and their parameters. The 3PL model is used in this and will be used in future trials.

In the trials it is observed, that students are surprised by the new evaluation methodology. They do not understand how their grade is calculated, since they are used to calculate it based on correct answers over total questions. To clarify the situation, professors explain to them how the new evaluation methodology works, and how it calculates grades.

This trial reveals some issues with the test taking module; some of them are technical problems and other are problems of the model. Technical issues include network problems and some minor inconveniences in the test taking interface; the test taking module is modified in order to deal with these technical issues. Additionally, results and complaints of students are reviewed. They revealed three issues with the Basic model:

- If from the beginning of the test a student answers four consecutive questions incorrectly (or correctly), the test is over and the student’s grade is 0 (or 5 respectively).
- In some cases the student ability got stuck after the second question.
- In some cases students answered a question correctly and the ability got lower.

These three issues are analyzed and its causes are discovered. The first issue is caused by the difficulty parameter. Professors set this parameter with integer values (e.g 1,2,...,7); this means that every time a student answers a question, his or her next question has a difficulty level increase or decrease by one unit. Given that the test starts with a question with medium difficulty, when a student fails three consecutive questions, he or she reaches the lower difficulty level; with his or her fourth incorrect answer, the difficulty level is out of the scale and the test is finished. The same situation happens when there are four consecutive correct answers: the difficulty level is out of the scale in its upper bound. Particularly, only students with a 0 grade complained about this situation.
The second and third issues are related to the discrimination parameter. A revision and analysis of the problematic question is done, and it revealed that some professors created questions with a low difficulty value and a high discrimination value. If a student fails a question with these parameters, the maximum likelihood function “gets stuck”, and the future answers of the student do not matter. The maximum value of the maximum likelihood function has the same \( \theta \) (ability) value.

Figure 3.6 and Table 2.2 show the general performance of this trial: 69% of students failed the test, 30% passed it with a grade high than 3, and 1% passed it with a grade equal to 3. Additionally, 60 students obtained a final grade of 0, and 72 students had a grade of 5.

![Figure 3.6. General performance of first test](image)

<table>
<thead>
<tr>
<th>Students</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>518</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>10</td>
<td>= 3</td>
</tr>
<tr>
<td>223</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>60</td>
<td>= 0</td>
</tr>
<tr>
<td>72</td>
<td>= 5</td>
</tr>
</tbody>
</table>

Table 3.2. General grades of first test (2013-I)

The platform administration module provides a tool that shows in detail the progress of students in a particular test. Professors use this tool in order to show to students their test information, including their grade, the number of questions taken and a graph that shows their progress in the test. Figures 2.8, 2.9, 2.7, 2.10 and 2.11 show some examples of students in this trial. Red points represent the answer of a question, 1 if it is correct and 0 if it is incorrect. The green line represents the ability of the student across the test and blue dots represent the difficulty of a question.

Figure 3.7 presents an example of the regular development of the test. The student starts with an item near medium difficulty, answers it correctly, and so, the second one is more difficult; he then answers incorrectly, and so, the third one is easier, and the process continues that way until the test ends. This is an example of an average student that answers 10 questions and has a grade of 3.33.

Figure 2.8 presents an example of the regular progression of the test where a student obtains a grade of 3.9, it has a little irregularity between questions 6 and 7 when the ability drops suddenly and difficulty of the questions is the same.

Figure 2.9 shows a regular progression of the test where a student obtains a grade of 5. The process of the student is very clear: he correctly answers the first three questions, and fails in the fourth one; after that he answers correctly and incorrectly some questions, and finally answers one correctly in the upper bound of the scale. Hence, the test ends.
Figure 3.7. Progress of a student in the first test (I)

Figure 3.8. Progress of a student in the first test (II)

Figure 2.10 shows how the test ends after four consecutive correct answers. Students that answer four questions correctly (starting from the first one), have a grade of 5; they finished the test on the average of a half hour. A similar situation happens when students answer four questions incorrectly. Students that answer incorrectly the first four questions obtain a grade of 0. Averagely, they answers the test in 20 minutes.

Figure 2.11 shows a special case in the progression of a test. This and some other cases are analyzed revealing that the main problem is a series of questions with a low difficulty and a high discrimination. This particular case modifies the regular behaviour of the
likelihood function, which creates irregularities in the ability estimation. For instance, in this case, ability reaches values below 1, something that should not happen.

### 3.4 Question History Model

In order to deal with the set of issues found in the Basic Model, some improvements that generate a second model called Question History Model are proposed. This model takes Basic Model structure and adds to it a history of questions. This history is used to decide if the difficulty of the next question will change or not (if it uses the adaptive process or
not). When a test starts, the history record initiates empty; when a student answers a question, the system stores the score of the question (1 if its correct or 0 if its incorrect) and presents a question of the same difficulty of the previous one. A search of questions of the same difficulty level is carried out in the item bank within a range close to the difficulty of last one answered. The system presents questions of the same difficulty until the history has 2 out of 3 questions correct or incorrect. When the history has the 2 out of the 3 questions required, the adaptive selection process of the Basic Model is used, and the history record empties. Figure 2.12 shows the structure of the Question History Model; the blue boxes show the components of the Basic Model and the red boxes show the improvements of this model.

In order to tackle previous issues with the discrimination parameter, the irregularities and its causes are explained to the professors. The effect of low difficulty questions with a high discrimination is highlighted. Particularly, this configuration is illogical since an easy question does not establish a significant difference between the knowledge of two students, therefore, this configuration should not be valid. To guarantee that this configuration is not set in the new trial or in future ones, the question creation interface is modified including a validation over the discrimination value.

This model is tested with the second exam of the computer programming course in the first semester of 2013 (same academic period of the previous trial). For this trial, a new item bank is created. The improvements integrate in this model can be seen in the progression of students. Figure 2.13 shows an example of the progression of a student in the trial of this model. The student starts the test with a medium level question and the history record starts empty ($qh = [0]$); then he answers the first question correctly; hence, this value is stored in the historical $qh = [1]$; after that, the system presents a question of the same difficulty level (medium); the second question is answered incorrectly, then the historical stores the answer, $qh = [1, 0]$; and then, the system selects a question with the same difficulty level again; the third question is answered correctly, then $qh = [1, 0, 0]$; at
this point, the system uses the adaptive selection and empties the history record $qh = [\emptyset]$. This process continues until the test reaches its stopping rule.

![Question History Model](image)

**Figure 3.12. Question History Model**

<table>
<thead>
<tr>
<th>Grade:</th>
<th>3.1667</th>
<th>Number of questions:</th>
<th>10</th>
</tr>
</thead>
</table>

![Progress of a student in the second test](image)

**Figure 3.13. Progress of a student in the second test**

Figure 2.14 and Table 2.3 show the general performance of the trial of the Question History Model. A 31% of the students passed the test with a grade higher that 3; 6% passed it with a grade equal to 3 and 63% failed the test. Additionally, almost 100 students did not take the test because they canceled the course. In general, statistics are similar to the first trial (trial of the Basic Model). However, the number of students that got a grade equal to 0 and 5 decreased.
Individual and general results of the test are analyzed, and it can be observed that the question history record helps to balance the test. Additionally, in the development of the trial, it can be observed that changes introduced in this model make students feel more comfortable with the evaluation process. Moreover, students now have previous experience with the evaluation methodology and understand it better; in this trial, a few students complained about their grade, but professors explained to them their exact progress and their doubts are solved.

After the test, students pointed out an issue that this model does not take into account. They said that they studied all subjects for the test, but only answered questions of a few of them, in some cases only one. This issue is discussed with the professors of the course and a solution to it is presented in the next version of the model.

![Figure 3.14. General performance of second test (2013-I)](image)

<table>
<thead>
<tr>
<th>Students</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>421</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>38</td>
<td>= 3</td>
</tr>
<tr>
<td>204</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>23</td>
<td>= 0</td>
</tr>
<tr>
<td>5</td>
<td>= 5</td>
</tr>
</tbody>
</table>

### Table 3.3. General grades of second test (2013-I)

#### 3.5 Topic History Model

The issue that is discovered in the previous model is considered and some changes are made in order to deal with it. The idea of the historical array that is introduced in the previous model, is used to create a topic history record. This is used to search for and show questions of all topics within a test. Figure 2.15 shows the improvements introduce in this model in the green boxes. These changes modify in the selection stage, when a student answers a question, the topic of that question is stored in a historical called \( th \). When the system looks a question, it searches for one of a topic that is not in the array \( th \). The array \( th \) is used in both selection methods, adaptive and same difficulty. If the algorithm does not find a question with different topic of the \( th \) content, it removes the first item in \( th \) and searches again. When a student takes questions of all topics, \( th \) is emptied.

In 2013-I period due to external reasons (social protests) the term was not carried on in a regular way. This interfered with the third test of the computer programming course in that academic period. Therefore, this model is tested in the next period (2013-II). The 2013-II period had three tests that are used as trials for this version of the model. With a new academic period, new students take the course and test this model, even though, a few students repeat the course.
Results from the first test of 2013-II period are shown in Figure 2.16 and in Table 2.4. They show that a 58% of the students passed the test with a grade higher than 3; 5% passed it with a grade equal to 3 and 37% failed the test. These results show an increase on the percentage of students that passed the test, compared to the previous trials (first and second test of 2013-I). The increase of students that passed the test is given by three circumstances:

- The model is progressively upgraded to cover the requirements and suggestions of students and professors.
- The creation process of the item bank is easier every time; professors have learned how to establish the parameters more accurately every time.
- Students are getting used to the evaluation method. They speak with other students (from previous semesters) about how the exam will be. Additionally, some students that cancel the course during a semester, take it again the next academic period and share their experiences with other students.

Some students that took the first test did not take the second one, and some that took the second did not take the third in the 2013-II period (similar to the 2013-I). Hence, the number of students in the trials decreased in almost 200 from the first to the third trial.

Results from the second test in the 2013-II period are shown in Figure 2.17 and in Table 2.5. In this trial, 51% of students passed the test with a grade higher than 3; 6%
passed it with a grade equal to 3, and 43% failed the test. Results from the third test in 2013-II period have a similar behavior. Figure 3.18 and Table 2.6 show the following results for the third trial: 55% of students passed the test with a grade higher than 3; 6% passed with a grade equal to 3, and 39% failed the test. In both cases, students with a grade of 0 are less than in the previous period (2013-I), and the percentage of students with grade of 5 increased.

Results from the second and third test of 2013-II, show a similar behavior to each other and to the first test. The general results of students are similar from the three test in 2013-II, which indicates a consistent behavior of the model. The general grades of students, shows that between 57% and 63% of students pass tests in 2013-II. These results show a general improvement of grades of students compared to the 2013-I period. Additionally, the number of students with a final grade of 0 is lower than tests in 2013-I. Hence, the addition of the topic history improves the model.

<table>
<thead>
<tr>
<th>Students</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>296</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>37</td>
<td>= 3</td>
</tr>
<tr>
<td>454</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>26</td>
<td>= 0</td>
</tr>
<tr>
<td>17</td>
<td>= 5</td>
</tr>
</tbody>
</table>

Table 3.4. General grades of first test (2013-II)

<table>
<thead>
<tr>
<th>Students</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>253</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>34</td>
<td>= 3</td>
</tr>
<tr>
<td>297</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>8</td>
<td>= 0</td>
</tr>
<tr>
<td>66</td>
<td>= 5</td>
</tr>
</tbody>
</table>

Table 3.5. General grades of second test (2013-II)
CHAPTER 3. COMPUTER ADAPTIVE TESTS PROCESS IN AN E-LEARNING PLATFORM

Figure 3.18. General performance of third test (2013-II)

<table>
<thead>
<tr>
<th>Students</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>163</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>24</td>
<td>= 3</td>
</tr>
<tr>
<td>230</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>4</td>
<td>= 0</td>
</tr>
<tr>
<td>52</td>
<td>= 5</td>
</tr>
</tbody>
</table>

Table 3.6. General grades of third test (2013-II)

3.6 Summary

This chapter presented the designing and building of a Computer Adaptive Test Model. It is integrated into the Intelligent E-Learning Platform of the Universidad Nacional de Colombia and the Computer Programming course is to proof the concept of the platform. It allows the easy use of the model in future versions of the course or in other courses. The proposed CAT model includes 1PL, 2PL and 3PL models from Item Response Theory in its selection process.

The development of the model is presented including its three versions: Basic Model, Question History Model and Topic History Model. The different versions of the model are created by using feedback from students and professors in different trials. The final version of the model includes two historical arrays that tackle some key issues found in the trials. The first array (question historical record) makes the difficulty change slower, this is done by choosing between two selection methods: the selection of an item with the same difficulty of the previous one, or the adaptive selection method. Additionally, the question history prevents that the tests ends when four consecutive questions are answered correct or incorrect (starting from the first one). The second array (topic historical record) guarantees that all topics in a test are evaluated. Results show that these changes improve the general results of students in tests.

The Computer Programming course is an example of how a course evaluation methodology can be changed from a standard evaluation methodology to an adaptive one. Additionally, it is important to point out that students and professors take some time to adjust to the new evaluation methodology.

The creation of item banks is done by using the experts approach. This approach allow courses that do not have a item bank, create one in a short period of time, for instance two week are required to create the item bank for the Computer Programming course.

Part of this chapter will be published in an article entitled “Evolution of Teaching and Evaluation Methodologies: The Experience in the Computer Programming Course at Universidad Nacional de Colombia” in the Revista de Ingeniería e Investigación de la Universidad Nacional de Colombia (Journal of Engineering Research of the National Uni-
versity of Colombia) [22]. Moreover, the platform development is published in two articles, the first one entitled “A Didactic E-Learning Platform with Open Content Navigation and Adaptive Exercises” in the proceedings of the International Conference on Education and E-Learning Innovations (ICEELI) [24], and the second one entitled “Plataforma de Aprendizaje y Cursos Masivos Abiertos en Línea de la Universidad Nacional de Colombia” (in english, “Learning Platform and Massive Open Online Courses at Universidad Nacional de Colombia”) in the proceedings of Virtual Educa 2013 [23].
Conclusions

• To pass from a standard evaluation methodology to a Computer Adaptive Test methodology, is not an easy transition for students and professors. In the establishment process of the proposed model, several key factors that influence this transition are observed. In the first place, students are surprised for the new evaluation methodology, for instance, they are not used to do not knowing how many questions has a test. Additionally, they have strong reactions when the test suddenly ends, especially when they answered a few questions. Students are surprised when a test ends after that they answered only four questions, students that obtained a grade of 5 react happily, however students that obtained a grade of 0 do not. In all the trials made, it is observed that students feel more comfortable when they answered more questions, in this way they think that the test is more fair. Hence, a good Computer Adaptive Test model has to consider this situation and has to make students feel fairly evaluated. In order to accomplish that, the proposed model includes two history records, one for questions and another for topics. These two history records are added to the model in order to solve some issues that are found during the trials.

• One of the most discussed issues of the Computer Adaptive Tests is the item bank. The process of obtain or create a perfect item bank is not an easy task, in order to obtain a item bank of the highest quality, a lot of time and resources have to be invest. The traditional item calibration requires that a high number of students take several trial tests in order to appropriately calibrate the items, and then, include the items in the item bank to be used in the real tests. Meanwhile, using experts approach (with professors of the computer programming course) a item bank for a test can be created in two weeks. The result of this process is a suitable item bank in a relatively short period of time. The use of this approach is an appropriate option for item bank creation at a University where various tests are done in a academic period.

• Due to each teacher has his or her own teaching methods, manage, coordinate and put professors into agreement about the item parameters is not easy; since, each professor has its own way to teach and to create questions. After some trials, the item bank creation process is easier and the review of questions is made efficiently. The item bank creation process still has some details to fix, but is a good starting point to using Computer Adaptive Tests as a evaluation method.

• The results of the clustering process, provides three clusters per trial, for all five tests. The number of clusters is defined through an exploration process using values from $k = 2$ to $k = 10$ in k-means. After the clustering process, a description process
that assigns a label to each cluster is done. These labels classify students within a cluster according to their answers. The labels have the following possible values: low-medium, medium-low, medium, medium-high and high-medium. The labels close to the medium ability appear frequently, these is consist with the exploration process carried out before the clustering process. Hence the results of the clustering process are consistent with the provided data.

- The trials made include tests in two academic periods (2013-I and 2013-II). Within each academic period, the resulting clusters are analysed in order to establish how many students remain in the same cluster, i.e., cluster with the same description in both tests, and how many change from one cluster to another. This analysis shows that a significant number of students remain in their initial cluster, hence the results over an academic period are consistent. Additionally, some students change from their initial cluster to another, a significant number of students change to a cluster with a high description value, in this way it can be observed an increase in the general performance of students.

- The results from the association rules process, shows several relations between questions. For instance, $P_{157} \implies P_{158}$, which is interpreted as follows: if a student answers correctly the question $P_{157}$, it is probable that he or she will correctly answer question $P_{158}$. These relations provide some information about questions but it is not clear enough. Hence, they are analyzed by considering the question content, this analysis reveals relations that provide more useful information. The analysis shows that these new relations represents the dependencies between topics of the course, e.g the topic matrix depends on the topic arrays. These results can be used to evaluate the structure of the course, to analyse if it the content is well distributed or if should be changed based on the students performance. Additionally, the relations between topics can be used by professors in order to improve their teaching methods and lessons by focusing in the dependencies of topics.

- The initial results of the association rules process, reveal a direct relations between two “tricky questions” in a test. According to a professor of the course students that fail these two questions do not read carefully and fall into its “trick”. The analysis shows that these two questions relate the topics of arrays and matrices (two dimensional arrays). The first question evaluate use of indexes through arrays and the second one, the definition of matrices (its dimensions). The relation discover said, that if students answer the first question correctly, it is probably that they probably answer the second one correctly; i.e. if students know how indexes are used in arrays, it is probably that they understand how an array (two dimensional) is defined. The discovery of this relation is very important, because illustrated that the association process is capable of find out specific relations between questions.

- The use of the Computer Programming Course as a proof of concept of the platform and the model is a success. The feedback of students and professors of the course provides fundamental information in the model creation. By integrating the Computer Adaptive Test model into the Intelligent E-Learning platform, it is allowed that future versions of the course or other courses have an easy access to this evaluation methodology.
Future Work

Although this Thesis has shown that Computer Adaptive Tests is a good evaluation method which can be applied to a regular course with the collaboration of professors, there are some aspects that can be addressed in future works:

- The item bank creation process that uses an approach based on experts, generates a good-enough bank that can be used in a Computer Adaptive Test. In addition to this, the initial data mining analysis used in this Thesis can be used in future researches in order to propose a model that identifies what items can be refined and how to adjust its parameters.

- A further study could use the information obtained of the data mining process, i.e., users profiles, as feedback for tests. For instance, this information could be used to select the initial item of a test based on previous results of students. The current work presents a Computer Adaptive Test model that considers each test as an independent object, each test starts with no previous information of the students. The addition of previous information of students to select the initial item of tests could improve the model proposed in this Thesis.

- Further research might explore the proposed dissimilarity measure, and propose new measures in order to create a group of measures to compare students. Such research would explore different characteristics of the students and questions in order to generate new dissimilarity measures.
Bibliography


[34] Unión Temporal E learning Colombia.


[61] Sena Virtual, online: http://www.senavirtual.edu.co/.


[66] Universidad Nacional Abierta y a Distancia, online: http://www.unad.edu.co/.

