

<http://www.lsi.us.es/> trinidad

Publicado en febrero 2026 por

Ismael Carrasco Mkhazni

Copyright © MMXXVI

ismcarmkh@alum.us.es

Pon aquí cuestiones acerca del copyright

Yo, 15456902M Ismael Carrasco Mkhazni con NIF número

DECLARO

mi autoría del trabajo que se presenta en la memoria de este trabajo fin de grado que tiene por título:

King And Peasants

Lo cual firmo,

Fdo. 15456902M Ismael Carrasco Mkhazni
en la Universidad de Sevilla
17/02/2026

Tu dedicatoria aquí



No olvides añadir una nota de agradecimiento a quienes hayan contribuido emocionalmente al proyecto fin de Grado.

Para una generación entera, la puerta de entrada al entretenimiento digital no requería potentes consolas ni largas instalaciones, sino simplemente una ventana del navegador. La infancia del autor estuvo definida por la inmediatez y accesibilidad de los juegos web, donde la diversión, ya fuera en solitario o compartida en línea, estaba a un solo clic de distancia. Sin embargo, con la obsolescencia de tecnologías como Flash, esa simplicidad se ha diluido en favor de ecosistemas más cerrados y complejos.

Este Trabajo Fin de Grado, titulado King and Peasant, nace de la motivación personal de recuperar esa esencia nostálgica, reconstruyendo la experiencia del juego de navegador clásico bajo los estándares de la ingeniería de software moderna. El proyecto presenta una plataforma multijugador de cartas en tiempo real desarrollada con una arquitectura MERN (MariaDB, Express, React, Node.js) y WebSockets. Más allá de la mecánica del juego, el sistema pone el foco en la interacción social que caracterizaba a aquellas comunidades, implementando búsqueda de usuarios, listas de amigos y salas de espera (lobbies) dinámicas. Todo ello, desplegado mediante contenedores Docker, demuestra cómo las tecnologías web actuales permiten revivir la magia del pasado con la seguridad, escalabilidad y rendimiento del presente.

For a whole generation, the gateway to digital entertainment did not require powerful consoles or lengthy installations, but simply a browser window. The author's childhood was defined by the immediacy and accessibility of web games, where fun, whether solo or shared online, was just a click away. However, with the obsolescence of technologies like Flash, that simplicity has given way to more closed and complex ecosystems.

This Final Degree Project, titled King and Peasant, is born from the personal motivation to recover that nostalgic essence, reconstructing the experience of classic browser gaming under the standards of modern software engineering. The project presents a real-time multiplayer card platform developed with a MERN architecture (MariaDB, Express, React, Node.js) and WebSockets. Beyond the game mechanics, the system focuses on the social interaction that characterized those communities, implementing user search, friend lists, and dynamic waiting rooms (lobbies). All of this, deployed through Docker containers, demonstrates how current web technologies allow us to revive the magic of the past with the security, scalability, and performance of the present.

X

2

The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck of his whole damn life – and one is as good as the other.

3

4

Ernest Hemingway (1899–1961),

5

Novelist

6

7

Resumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este capítulo?

8

9

1.1 EL MUNDO DEL X (VIDEOJUEGO, E-COMMERCE,...)

10

Hay que ir poco a poco acotando el contexto donde se desarrolla el proyecto. No se debe sobreentender que el evaluador de la memoria sabe del tema. Escribid el texto para la abuela.

11

12

13

1.2 SUBCONTEXTO

14

1.3 SUBSUBCONTEXTO

15

1.4 ESTADO DEL ARTE

16

Cómo se encuentra la industria hoy en día a nivel económico y tecnológico.

17

18

*The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck
of his whole damn life – and one is as good as the other.*

19

20

Ernest Hemingway (1899–1961), 21

Novelist 22

23

A *qui mal un breve resumen del capítulo.*

24

2.1 MOTIVACIÓN

25

Esta sección se rellenará cuando tengamos un producto de mercado en lugar de un proyecto en el que haya un cliente específico. Deberá justificar brevemente el problema a resolver, escenario en el que se aplica, hipótesis de partida, público objetivo, etc.

26

27

28

2.2 LISTADO DE OBJETIVOS

29

Objetivo 1. Blabla Detalles del objetivo 1.

30

Objetivo 2. Blabla Detalles del objetivo 2.

31

33

*The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck
of his whole damn life – and one is as good as the other.*

34

35

Ernest Hemingway (1899–1961),

36

Novelist

37

38

Resumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este
capítulo?

39

40

3.1 ESTRUCTURA ORGANIZACIONAL DEL PROYECTO

41

¿Se hace en grupo? En caso afirmativo, ¿cuál va a ser la responsabilidad de cada uno?

42

43

3.2 METODOLOGÍA DE DESARROLLO

44

Indicar en qué metodología nos basamos, explicarla brevemente y luego adaptarla a nuestras necesidades. Cada una de estas cuestiones debe ser una subsección.

45

46

47

The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck of his whole damn life – and one is as good as the other.

48

49

Ernest Hemingway (1899–1961),

50

Novelist

51

52

Resumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este capítulo?

53

54

4.1 RESUMEN TEMPORAL DEL PROYECTO

55

Resumen del proyecto	
Fecha de inicio	10/10/2014
Fecha de fin	10/10/2014
Periodicidad de las revisiones	3 semanas
Carga de trabajo semanal	12 horas
Horas totales previstas	225 horas
Horas finales	234 horas

Cuadro 4.1: Tabla resumen de tiempos y planificación

4.2 PLANIFICACIÓN INICIAL

56

Aquí un desglose de las iteraciones, comienzo y fin de cada una:

57

Resumen de iteraciones	
Iteración 1	10/10/14 a 21/10/14
Iteración 2	21/10/14 a 15/11/14
...	dd/mm/aa a dd/mm/aa

Cuadro 4.2: Planificación temporal de iteraciones

Explicar cómo se han decidido las fechas, interacción con fechas importantes y situaciones personales.

58

59

ESTE CAPÍTULO DEBE ESCRIBIRSE AL COMIENZO DEL PROYECTO

60

4.3 INFORME DE TIEMPOS DEL PROYECTO

61

Lo mismo que el anterior pero con datos reales. Ver Tabla ??.

62

Justificar los retrasos de forma detallada aquí para cada una de las iteraciones. Explicar las razones.

63

64

Resumen de iteraciones

Iteración 1 10/10/14 a 21/10/14

Iteración 2 21/10/14 a 15/11/14

... dd/mm/aa a dd/mm/aa

Cuadro 4.3: Planificación temporal de iteraciones

65

*The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck
of his whole damn life – and one is as good as the other.*

66

Ernest Hemingway (1899–1961), 68

Novelist 69

70

Resumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este capítulo?

71

72

5.1 RESUMEN DE COSTES DEL PROYECTO

73

Resumen del proyecto	
Costes de personal	5.045 €
Sueldo neto	2.030 €
Impuestos	1.000 €
Costes sociales	2.015 €
Costes materiales	560 €
Costes indirectos	450 €
TOTAL	8.000 €

Cuadro 5.1: Tabla resumen de costes

5.2 COSTES DE PERSONAL

74

Ya hablaremos de esto

75

5.3 COSTES MATERIALES

76

Y de esto también. Ver Sección ??.

77

5.4 COSTES INDIRECTOS

78

Y esto es una fiesta

79

81

The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck of his whole damn life – and one is as good as the other.

82

83

Ernest Hemingway (1899–1961),

Novelist

84

85

86

Resumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este capítulo?

87

88

6.1 LISTA DE CARACTERÍSTICAS

89

Aplicar aquí la primera iteración de Feature Driven Development.

90

6.2 DISEÑO ARQUITECTÓNICO

91

Descripción de los sistemas de producción, preproducción y pruebas.

92

93

The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck of his whole damn life – and one is as good as the other. 94
95

Ernest Hemingway (1899–1961), 96

Novelist 97

98

Resumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este capítulo? 99
100

7.1 CARACTERÍSTICAS A DESARROLLAR

101

1. Funcionalidad A. Ver Tabla ??.

102

2. Funcionalidad B.

103

Análisis de valor aportado 0001

Propuesta	Trabajo que pretende analizarse y justificarse
Valor	Qué valor aporta al proyecto o al usuario final.
Coste	Qué costes en términos de esfuerzo, adquisiciones y limitaciones tiene la propuesta
Opciones	Qué otras opciones se tienen que aporten un valor similar? ¿Es realmente un valor relevante para el proyecto/cliente
Riesgos	Qué riesgos pueden surgir a la hora de desarrollar esta propuesta.
Deuda técnica	Posibles deudas técnicas que se asumen con el desarrollo de esta propuesta.

Cuadro 7.1: Análisis de valor aportado 0001

7.2 DISEÑO

104

Aquí una discusión de cómo va a afectar todo al diseño

105

Debe insertarse un diagrama UML de diseño con los cambios y hacer referencia en el texto así Fig. ??.

106

107

Un memorando técnico por cada decisión de diseño.

108

7.3 IMPLEMENTACIÓN

109

22

Un memorando técnico por cada decisión de implementación y refactorización que afecte al diseño.

110

111

Figura
pendiente

Aquí el modelo de diseño en formato vectorial preferentemente (pdf)

Figura 7.1: Diagrama UML de diseño para la iteración 1

Memorando técnico 0001

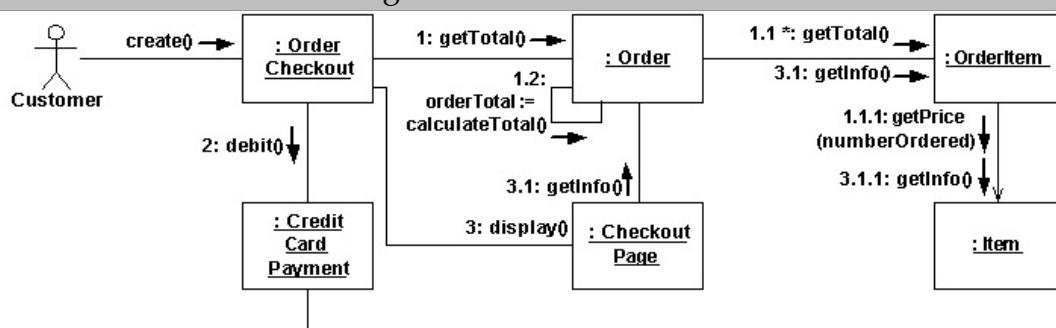
Asunto	¿Cuál es el problema?
Resumen	¿Cuál es la solución propuesta?
Factores causantes	Descripción pormenorizada del problema
Solución	Descripción pormenorizada de la solución propuesta
Motivación	¿Por qué propone esta solución?
Cuestiones abiertas	Factores a tener en cuenta en la solución cuya dimensión se reconoce.
Alternativas	Otras soluciones consideradas y la razón por la que se excluyeron.

Cuadro 7.2: Memorando técnico 0001

Identificador	Descripción de la acción de alto nivel	
0001	Prueba	
Métodos de alto nivel		
[return_type] method_name1 (param1:type1, ...)		
Pasos (Usar Pseudocódigo o similar)		
1. Paso 1.		
2. Paso 2.		
Métodos de bajo nivel necesarios		

Paso	Clase	Método	Mem.	IU
			Técn.	
1	ClassName	[return_type] me	001	GT

Diagrama de Colaboración



Identificador		Descripción de la acción de alto nivel		
alvotermar02		Grubber		
Métodos de alto nivel				
[return_type] grubber (param1:type1, ...)				
Pasos (Usar Pseudocódigo o similar)				
<p>1. Lanzar 2 dados</p> <p>2. Compara resultado de los dados con kicking del open-side</p> <p>2.1. Si valor dados es menor o igual a kicking, avanza 10m</p> <p>3.1. Si no hay defensa y el golpeo es exitoso, el pateador retiene la posesión del balón</p> <p>3.2. Si hay defensa y el golpe es exitoso, el atacante tira un dado y suma su valor al de speed y strength y el defensor lanza 2 dados y lo suma al valor de speed y strength de su jugador, el vencedor será aquel que tenga más puntos, si es igual, la posesión es del defensor</p> <p>4.1. Si no es exitoso y hay defensa el balón pasa a posesión del defensor</p> <p>4.2. Si no es exitoso y no hay defensa de lanza un line-out</p>				
Métodos de bajo nivel necesarios				
Paso	Clase	Método	Mem.	IU
			Técn.	
1	Dice	[Integer] throwDice ()	001	SI
2	ClassName	[Int] compareKickingToDice (kicking:Integer, dice: Integer)	001	SI
2.1	ClassName	[Integer] setLine (line:Integer)	001	SI
4.2	ClassName	[Integer] lineOut ()	001	SI

115

7.4 PRUEBAS

116

Descripción de las pruebas realizadas al software

117

7.5 DESPLIEGUE

118

Breve resumen de cómo se han desplegado los cambios en el sistema de producción.

119

122

*The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck
of his whole damn life – and one is as good as the other.*

123

124

Ernest Hemingway (1899–1961), 125

Novelist 126

127

Resumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este
capítulo?

128

129

8.1 SECCIÓN LIBRE

130

Estructurar en función del proyecto.

131

132

The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck of his whole damn life – and one is as good as the other.

133

134

Ernest Hemingway (1899–1961),

Novelist

135

136

137

Resumen de lo que va a ocurrir en el capítulo. ¿Cuál es el objetivo que tenemos con este capítulo?

138

139

9.1 INFORME POST-MORTEM	140
Qué es un informe post-mortem	141
9.1.1 Lo que ha ido bien	142
■ Argumento a favor 1.	143
■ Argumento a favor 2.	144
■ Argumento a favor 3.	145
9.1.2 Lo que ha ido mal	146
■ Argumento en contra 1.	147
■ Argumento en contra 2.	148
■ Argumento en contra 3.	149
9.1.3 Discusión	150
En función de lo anterior, qué cambiaría si empezara hoy el proyecto de nuevo.	151
9.2 TRABAJOS FUTUROS	152
Enumera los puntos abiertos y que no se han resuelto. Indica si darían lugar a otro proyecto y de qué forma se podría acotar.	153
	154

156

*The good parts of a book may be only something a writer is lucky enough to overhear or it may be the wreck
of his whole damn life – and one is as good as the other.*

157

Ernest Hemingway (1899–1961), 159

Novelist 160

161

This is an example of an abstract. Multiple lines are supported. Several paragraphs. It 162
jumps to the next page. Blau blau blau. I am introducing more text to reach the third 163
line 164

A.1 SOFTWARE PRODUCT LINES

165

- Objective of a *Product Line (PL)* (mass production and customisation) [?] 166
- The focus in software derives in *Software Product Lines (SPLs)*. 167
- Variability management: variability models 168
- When and how are used VMs: FMs are described in FODA report as a key element in SPL since they represent the variability and commonality of the different products in a SPL. 169
170
171

A.2 FEATURE MODELS

172

To Abductive Section in 2.1

173

As the number of products to be built by a SPL may be large and the constraints among features may be complex, representing such an information in a manageable and compact manner is a must. *Feature Models (FMs)* represent the set of products a SPL may build in terms of product features. Some features are optional while others are mandatory. To indicate the relationships among features, they are hierarchically linked, forming a tree whose root is a feature representing the whole functionality of a product. The root feature is refined in child features, which increase the level of detail and reduce the scope of features. Recursively following this refinement process, a tree-like structure is obtained where three basic kinds of hierarchical relationships are used:

174

175

176

177

178

179

180

181

182

- Mandatory: a mandatory relationship affects a parent and child feature. It forces the child feature to appear in a product whenever its parent feature does. 183
184
- Optional: a child feature connected to a parent feature by means of an optional relationship may be optionally selected whenever its parent feature is. 185
186
- Set-relationships: three or more features are part of a set-relationship: a parent feature and a set of two or more child features. A set-relationship contains a cardinality that constraints the number of child features to be selected in a product whenever its parent feature is selected. If the cardinality is [1..1] it is commonly remarked as an *alternative relationship* where only one child feature may be selected at the same time. If the cardinality is [1..N] (where N is the number of 187
188
189
34
190
191
192

child features), it is also known as an *or-relationship* as any combination of child 193
features is allowed while at least one is selected. 194

Although FMs can represent most of the most frequent constraints, the hierarchical 195
nature of these models might hinder the representation of some constraints. Under this 196
circumstance, *cross-tree constraints* can be added. The most common kinds of cross-tree 197
constraints are: 198

- Dependency: a feature depends on another feature if the second one must be part 199
of a product whenever first one is selected. 200
- Exclusion: two features exclude themselves if both of them cannot be part of a 201
product at the same time. 202

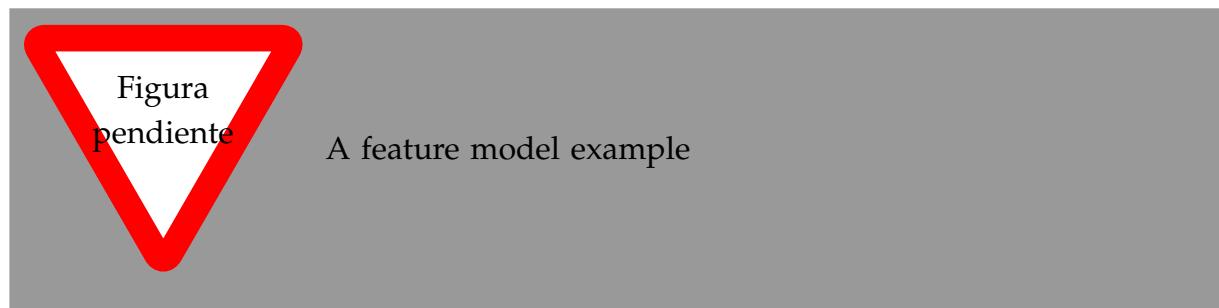


Figura A.1: An example of a Home Integration System

The example in Figure ?? describes a *Home Integration System* (HIS) SPL in terms 203
of its features and the relationships among them. Leaning on this example we define 204
some useful terms: 205

Partial configuration : a partial configuration is composed by three sets of selected 206
(S), removed(R) and undecided(U) features. A feature can only be in one of these 207
sets and every feature in the FM (fm) must be in one of them, i.e. $S \cup R \cup U = 208$
 fm and $S \cap R \cap U = \emptyset$. A partial configuration represents an intermediate state 209
during the process of a customer selecting the feature for a custom product. For 210
example, $S_P = \{ \dots \}$, $R_P = \{ \dots \}$ and $U_P = \{ \dots \}$ define a partial configuration for 211
the sample FM where some features are still to be decided if they are to be selected 212
or removed in a configuration. 213

(Full) configuration : a full configuration or simply a configuration is a partial config- 214
uration such that the set of undecided features is empty. For example, $S_F = \{ \dots \}$ 215
and $R_F = \{ \dots \}$ describe a full configuration for the example FM. 216

Product : a product is a representation for a full configuration such that only the selected features are remarked. For instance, $P = \{\}$ is a product for the above full configuration. A product such as A,B is a valid since all the constraints within the FM are satisfied. However, A,B and C is not a valid product since D is required. 217
218
219
220

Validation A partial configuration is *valid* if all the relationships and constraints are satisfied given the sets of selected, removed and undecided features. So the definition applies for valid full configurations and valid products. As a conclusion we can affirm that a FM represents all the valid products in a SPL. 221
222
223
224

Objetivo: Briefly expose attributes as an important asset in feature models. 225

It is frequent that features are not enough to represent information that is relevant to represent a SPL variability. In this case, FMs are extended with feature attributes such as cost, versions, RAM consumption, etc. in the so-called *Extended Feature Models (EFMs)* [?]. Besides relationships, an EFM contains constraints that affect attributes which reduce even more the set of products a FM describes. Above definitions remain when attributes are introduced into FMs. 226
227
228
229
230
231

A.3 AUTOMATED ANALYSIS OF FEATURE MODELS 232

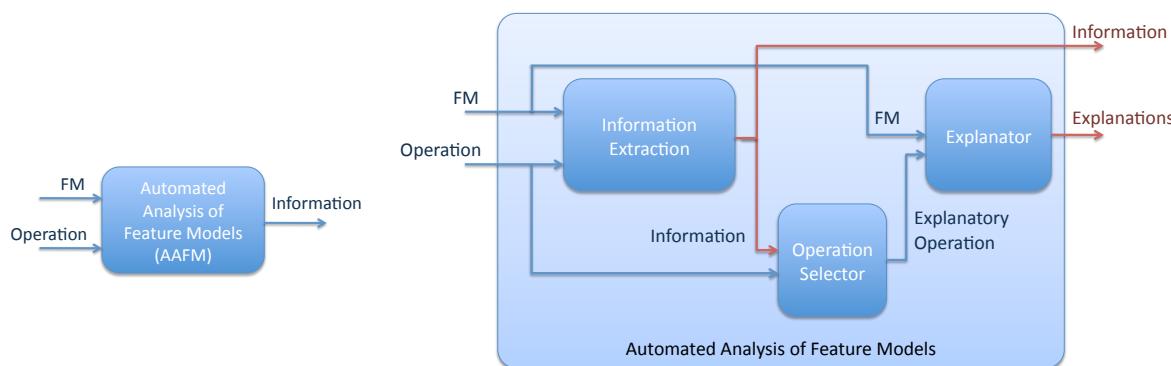
A.3.1 Scope 233

To Abductive Intro 234

FMs are used all along the SPL development as key models and many of the development decisions are taken relying on the information contained within them. Most of the times, relationships are complex and hinder the manual extraction of information. Manually obtaining information such as 'which is the product that costs the less?', 'does the feature model contain errors?' or 'why there exist no product containing certain features?' can be an unfeasible task. The complexity and compactness of FMs justify the need of an automated support of these operations. So the *Automated Analysis of Feature Models (AAFM)* arises as a topic of interest to deal with this problem in the SPL community. 235
236
237
238
239
240
241
242

The AAFM can be seen as a black-box process that receives a FM and an operation as inputs and obtains information (its kind depends on the analysis operation) as an 244
245

output (Fig. ??). There are many operations that extract information from a FM such as
 'counting products' operation whose result is a natural number indicating the number
 of customised products that can be built; or 'list of products' operation that obtains
 each of those products. This vision of AAFM as a black-box is valid for a subset of
 analysis operations that we call *information extraction operations* (IEO) that can be seen
 as processes to extract information from FMs. In other words, an IEO makes explicit an
 implicit information within a FM.



(a) The AAFM seen as a black-box process

(b) Extending the AAFM process with explanations

Figura A.2: A different view on AAFM distinguishing between information extraction and explanatory operations

Use me to explain
 in a larger text
 than 'sidetext'
 anything that is impor-
 tant to a reader not
 familiar with the
 dissertation context
 for example.

However, there is a subset of analysis operations known as *explanatory operations* (EO) whose objective is explaining the result obtained from a IEO. Sometimes, the result is not the expected one and the analyser needs to know which are the relationships that have caused it. For example, let us suppose that the IEO 'which are the products described in a FM that cost less than \$1000?' obtains no products as a result. If we were expecting to obtain at least one product, it is important to determine the relationships in the FM that are responsible of that behaviour, so an EO 'why there is no product costing less than \$1000?' will shed light on the relationships that avoid obtaining any product. Ob-

taining no result is not the only case that claims for explanations. If we obtained only one product as a result and we were expecting to obtain at least 10 products, although an answer is obtained the result is unexpected and the discrepancy reasons have to be found. Moreover, explanatory operations are also useful even when an expected result

is obtained, to reinforce the certainty that the result is correct. So it can be concluded 269
that EO^s complement the information an FM analyser obtains from IEO^s. 270

The complexity of feature modelling relies on correctly setting the relationships that 271
describe the set of products to be built in a SPL. Relationships are the only elements 272
responsible of the results obtained in FM analysis. So an *explanation* is a set of relation- 273
ships that may have caused that result. While IEO provides for an unique response 274
that is known for certain, an EO provides for a set of probable explanations to a result 275
obtained from a IEO, being only one of them a valid explanation. It would be the anal- 276
yser the one in charge of discriminating the correct explanation, maybe performing 277
new analysis operations. 278

**THIS IS A SIDE TEXT. USE TO
REMARK IMPORTANT
INFORMATION**

Therefore, two kinds of operations are distinguished 279
in AAFM: information extraction and explanatory oper- 280
ations. Explanatory operations have no sense without a 281
paired information extraction operation and its result. To 282
ensure that explanatory operations are always paired to 283
an information extraction operation, we define a new black- 284
box process of AAFM that incorporates explanations as 285

an additional output (see Figure ??) 286

1. Information extraction: the original process, which remains the same. 287
2. Operation selector: depending on the information extraction operation the anal- 288
yser asks for and the information obtained as a result, this process provides the 289
explanatory operation to be performed. In other words, it pairs an explanatory 290
operation to an information extraction operation. 291
3. Explanatory analysis: provides a set of explanations from the FM and the ex- 292
planatory operation. 293

The overall process can be encapsulated into a holistic black-box process which 294
receives the FM and the information extraction operation as inputs and provides a 295
result and explanations as outputs. It can be seen as we just add explanations as an 296
output to the analysis process. 297

To realise this view on the AAFM, we need to get 38 details on the insides of these 298
black-boxes. Since the information extraction process is already rigourously defined in 299
Benavides' PhD dissertation, the purpose of this paper is defining the remaining two 300

sub-processes. We formalise the explanatory analysis process by means of default logic 301
and provide the criteria to implement the operation selector process. 302

Most Common Techniques to perform AAFM Operations. 303

A.4 DYNAMIC SOFTWARE PRODUCT LINES (DSPL) 304

What is a *Dynamic Software Product Line (DSPL)*. Different points of view. What is 305
important is the automation of reconfiguration properties relying on SPL techniques. 306

We focus in the application of explanations in DSPLs as an application of our results. 307
Specifically we have worked in MAS and smart homes providing a solution for 308
automating product reconfiguration. 309

A.5 HYPOTHESIS AND OBJECTIVES 310

Objetivo: Justifying that explanations are a particular set of operations in AAFM
that are not solvable by means of the techniques that are used up-to-date

311

Objetivo: Set an impacting phrase that summarises the hypothesis

312

Hypothesis

*Explanations cannot be solved by AI techniques used to solve AAFM.
There should exist other AI techniques to solve explanations.*

313

Objective of the dissertation

*Defining a framework to provide solutions for explanatory analysis in
FMs.*

314

This dissertation summarises our contribution to solve some of the objectives we set in our PhD project. 315
316

- Defining a catalog of analysis operations where explanations are applied. 317
- Rigorously defining these operations in terms of logics. 318
- Proposing solutions to these operations. 319
- Validating our results by means of tools and projects where they are applied. 320

Next chapter focuses on refining how we have contributed to deal with the above objectives. 321
322

A piece of code... 323

```
public Map<Cardinality, CardinalValue> detectWrongCardinals() {  
    // any other implementation of Map can be used instead.  
    Map<Cardinality, CardinalValue> result =  
        new TreeMap<Cardinality, CardinalValue>();  
    for (r : relationships) {  
        if (r instanceof Set) {  
            Set set = (Set)r;  
            Cardinality card = set.getCardinality();  
            Domain dom = card.getDomain();  
            for (value: dom.getValues())  
                if (isWrongCardinal(card, value))  
                    result.put(card, value);  
        }  
    }  
    return result;  
}
```

A coolTable. Use inside a table. 342

Use \TableSubtitle{n,title} to add a subtitle as the header. n is the number of columns 343
and title is the text to place. [?] 344

A Catalog of FM Explanatory Operations (2009 version)		
Information Extraction Operation	FM Explanatory Operations	
	<i>Why? operation</i>	<i>Why not? operation</i>
Valid FM	-	invalid FM
Valid Configuration	valid partial conf.	invalid partial conf.
Valid Product	valid product	invalid product
Products Listing	vaild Product/Config	invalid FM/Product/Config
Products Counting	vaild Product/Config	invalid FM/Product/Config
Optimisation	vaild Product/Config	invalid FM/Product/Config
Core feature	core feature	core feature
Variant feature	variant feature	variant feature
Dead feature detection	-	dead feature
False-optional feature detection	-	false-optional feature
Wrong-cardinality detection	-	wrong cardinal
Information Extraction Operation	Configuration Explanatory Operations	
	<i>Why? operation</i>	<i>Why not? operation</i>
Valid Configuration	valid partial conf.	invalid partial conf.

Cuadro A.1: Most frequently used explanatory operations and their corresponding information extraction operations

-
- [] D. Benavides, A. Ruiz-Cortés, and P. Trinidad. Automated reasoning on feature models. *LNCS, Advanced Information Systems Engineering: 17th International Conference, CAiSE 2005*, 3520:491–503, 2005. ISSN 0302-9743. (pages 34, 36 y 40).