

Role Player Assistant: Entorn web per jocs de rol

Javier del Toro Vallina

Resum— Cada cop més trobem que la societat ha anat evolucionant fent ús de les noves tecnologies i aplicant-les a les situacions quotidianes. En un àmbit com és els del jocs de rol de taula, on cada cop podem observar a més persones utilitzant dispositius tecnològics per intentar incrementar l'experiència dels jugadors durant la partida, sempre limitant-se a eines que estiguin al seu abast. En el present projecte buscarem la manera d'aprofitar els nous dispositius mòbils que gran part de la població té al seu abast, per facilitar les interaccions que es porten a terme durant el transcurs d'una partida i alliberant als jugadors de certes limitacions que es presenten sense aquest dispositius. Per altre banda, continuant amb la línia del projecte i com a creixement personal, realitzarem una immersió en una nova tecnologia de desenvolupament, enfocada precisament a desenvolupar plataformes compatibles amb qualsevol tipus de dispositiu.

Paraules clau— Rol, fòrum, web, jocs, daus, tauler, dinàmic, dispositius mòbils.

Abstract— Often we find that society has been evolving using new technologies and applying them to everyday situations. In an area such as role playing games, where frequently we can see more people using technological devices trying to increase the experience of the players while they are playing, always limited to tools that they know how to use. In this project we will seek new ways to benefit from mobile devices which much of the population has to its disposal, to facilitate the interactions that take place during the course of a game and releasing players from some limitations that exist without these devices. On the other hand, continuing with the line of the project and as personal growth, we will learn a new development technology, focused specifically on developing platforms compatible with any device.

Index Terms— Rol, forum, web, game, dice, board, dynamic, mobile devices.



1 INTRODUCCIÓ

EL projecte que presentarem al llarg d'aquest document tractarà sobre el desenvolupament d'una aplicació web per donar suport a partides de rol de taula amb més dinamisme en un entorn de fòrum, mitjançant sales privades on els jugadors podrien arribar a disposar d'un tauler virtual, un chat, un llançador de daus de diferent tipus, fitxa de personatge, ajudes al director de joc.

El desenvolupament d'un projecte amb aquestes característiques sorgeix de l'aplicació de les noves tecnologies en un entorn analògic, per donar suport als jugadors mitjançant el seu dispositiu mòbil.

En aquest article podrem trobar els aspectes més importants treballats durant el transcurs del treball realitzat amb

motiu del final d'estudis de la menció d'Enginyeria del Software. Començarem per una presentació dels objectius que es volien assolir quan es va iniciar el treball, a continuació, veurem unes aplicacions similars a la que es vol desenvolupar i que ens van ajudar en el moment de buscar funcionalitats o determinar requisits. Un cop vist els objectius i aplicacions similars, explicarem les metodologies aplicades al llarg de les fases en les que es va dividir el treball, també detallarem aquestes i podrem trobar la planificació que es va realitzar per dur-les a terme. Gràcies a la realització d'aquestes tasques, obtenim uns requeriments pel projecte, dels quals veurem els més importants abans de detallar l'arquitectura de la plataforma, la qual, es mostrarà en un diagrama de classes, aquestes quedaran àmpliament explicades a l'apartat dedicat al desenvolupament.

Per últim, explicarem els comentaris obtinguts en un petit testeig realitzat a la plataforma amb usuaris reals, els resultats i les conclusions assolides amb aquest treball. Per acabar plantejarem una sèrie d'idees que poden permetre continuar aquest projecte.

-
- E-mail de contacte: javier.deltorovallina@gmail.com
 - Menció realitzada: Enginyeria del Software.
 - Treball tutoritzat per: Lluís Gesa Bote UAB (DCC)
 - Curs 2016/17

2 OBJECTIUS DEL PROJECTE

Aquest projecte ve motivat per la meva afició als jocs de rol, i la necessitat observada al llarg del temps d'implementar alguna aplicació per millorar l'experiència de joc adaptada a les tecnologies i les comoditats que ara tenim.

A continuació, presentarem els diferents tipus d'objectius a assolir en aquest projecte, primerament veurem un objectiu general, que engloba tot el projecte i després uns objectius més específics, que ens van ajudar en l'obtenció d'uns requeriments adients al nostre projecte.

2.1 Objectiu general

Desenvolupar una aplicació web responsive fen-la així accessible per dispositius portàtils, facilitant més interaccions per part dels usuaris, on incorporarem funcionalitats per poder realitzar partides de rol o assistir aquestes. Tot aquest projecte es desenvoluparà per suposat fent ús de metodologies pròpies de l'Enginyeria del Software.

2.2 Objectius específics

- Determinar quines funcionalitats poden necessitar els jugadors d'una partida.
- Determinar quines funcionalitats pot necessitar el director de joc d'una partida.
- Integrar l'aplicació web per dispositius mòbils.
- Realitzar una interfície d'aplicació senzilla i fàcil d'utilitzar.
- Permetre creació de personatge sense necessitat de partida.
- Donar autonomia a les partides sense necessitat de director de joc.
- Satisfer els requeriments especificats al document de requeriments.

2.3 Prioritat dels objectius

Un cop tenim clars els objectius del projecte, es va aplicar una prioritització sobre aquests per determinar la seva importància, la qual detallarem a continuació.

TAULA 1: LLISTAT D'OBJECTIUS I PRIORITATS

Objectius	Prioritat
Determinar quines funcionalitats poden necessitar els jugadors d'una partida.	Crític - Prioritari
Determinar quines funcionalitats pot necessitar el director de joc d'una partida.	Crític - Prioritari
Integrar l'aplicació web per dispositius mòbils.	Prioritari
Realitzar una interfície d'aplicació senzilla i fàcil d'utilitzar.	Secundari
Permetre creació de personatge sense necessitat de partida.	Secundari
Donar autonomia a les partides sense necessitat de director de joc.	Secundari
Satisfer els requeriments especificats al document de requeriments.	Crític - Prioritari

3 ESTAT DE L'ART

Entre les diferents aplicacions que poden existir amb funcionalitats similars, de les quals podem agafar idees o requisits no contemplats, trobem dues aplicacions que incorporen funcionalitats interessants, les quals podríem introduir o millorar en el nostre projecte.

3.1 Comunidadumbria.com

L'aplicació que ens presenten és www.comunidadumbria.com [1], una comunitat que desenvolupa una plataforma web on es poden jugar partides de rol dinàmiques amb un tauler virtual de forma gratuïta. Ens presenten una web elegant de fàcil accés per que qualsevol pugui gaudir de les seves funcionalitats des d'un ordinador. Realitzen partides de diversos jocs de rol mitjançant la seva pròpia aplicació amb ajudes al director de joc.

Donada l'experiència i el fet que desenvolupin la seva pròpia aplicació ens permet extreure funcionalitats i/o requeriments que no havíem pensat en primer moment, per altre banda no incorporen la possibilitat de jugar partides sense necessitat de director de joc, alhora que la web no es fàcilment utilitzable des de dispositius mòbils.

3.2 MUD Medina

MUD Medina [2], és una aplicació dedicada a narrar una història interactiva on l'usuari pren decisions i realitza accions per avançar en el joc. L'aplicació permet crear personatges al iniciar la partida i avançar en la història de forma automàtica mitjançant missatges a l'usuari.

Encara que en aquest cas l'aplicació no presenta l'aspecte social desitjat per aquest projecte, disposa d'una bona autonomia a l'hora de presentar l'història i les accions de l'usuari fet que pot aportar idees a l'hora de dissenyar el nostre sistema.

4 METODOLOGIA

Per les característiques que presenta el projecte diferenciam dos metodologies de treball:

- SCRUM.
- Desenvolupament per Característiques (Feature Driven Development - FDD) [3].

Hem fet ús de la metodologia SCRUM durant tot el transcurs del projecte per redactar la documentació necessària i a la fase de desenvolupament. Quan parlem d'aquesta metodologia ens referim a una adaptació, donat que no som un equip complet, així doncs hem aplicat les revisions de treball efectuat i les següents tasques a efectuar.

Desenvolupament per característiques és una metodologia àgil dissenyada pel desenvolupament de software, basada en la qualitat i el monitoratge constant del projecte, que ens ha permès assegurar que cada una de les característiques que hem anat desenvolupant estiguessin ben dissenyades i testejades, a més de permetre una fàcil modificació d'aquestes en cas que fos necessari.

A continuació es mostrarà un diagrama dels passos que hem realitzat a l'hora d'aplicar aquesta metodologia.

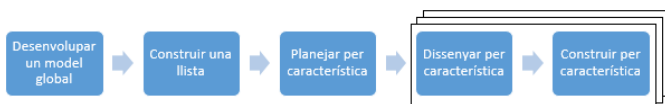


Fig 1: Diagrama FDD

5 FASES DEL PROJECTE

5.1 Fase de preparació

En aquesta fase, hem definit el que és important pel transcurs del projecte i que dóna informació per poder continuar amb la fase següent. També es va realitzar una primera planificació per aquest projecte. Aquesta fase es va realitzar fent ús de la metodologia "SCRUM".

5.2 Fase de desenvolupament

Durant aquesta segona fase hem anat desenvolupant la nostra aplicació fent ús de la metodologia "Feature Driven Development", que ens ha permès realitzar un procés iteratiu sobre el disseny, construcció i testeig de característiques, dividint així el projecte en petites tasques per tal d'obtenir funcionalitats, a més de la documentació necessària per continuar el seu desenvolupament en cas que fos possible.

5.3 Fase d'implantació

Tot i que no tindrà cabuda al nostre projecte, donada la falta de temps, aquesta fase es va planificar per obrir la plataforma al públic, donant per finalitzat el desenvolupament d'aquesta i passant a una fase de manteniment per arreglar errors o millorar alguna funcionalitat.

5.4 Activitats per fase

A continuació trobem una taula on podrem veure com es van dividir les tres fases abans mencionades amb una explicació del que es va realitzar en aquestes.

TAULA 2: ACTIVITATS DE CADA FASE

Fase	Activitat	Descripció
Fase de preparació	Planificació	Es va realitzar una primera planificació mitjançant el resultat d'un primer anàlisi, vam generar i estructurar les fases, activitats i tasques que compondrien el nostre projecte.
	Anàlisi	Aquesta activitat va tenir com objectiu realitzar una recerca del tema a desenvolupar per extreure informació rellevant, com trobar una tecnologia adient pel desenvolupament del projecte.
	Disseny i Prototip	En aquesta activitat es va començar a dissenyar les parts

		necessàries i generar la documentació adient per continuar amb la fase de desenvolupament.
Fase de desenvolupament	Implementació	En l'inici d'aquesta fase, mitjançant la documentació anterior, és on vam desenvolupar el codi que generaria la nostra aplicació, mitjançant iteracions que van donar com resultat les funcionalitats requerides per l'aplicació.
	Testeig Unitari	Es va donar accés a un grup reduït d'usuaris reals a l'aplicació per tal de detectar errors i retornar feedback.
Fase d'implantació	Testeig	Donarem accés a usuaris reals a l'aplicació per tal de detectar errors i retornar feedback.
	Manteniment	Aplicar correccions o noves funcionalitats a l'aplicació per tal de millorar-la.

6 PLANIFICACIÓ

A la taula següent podem veure com es van planificar les dates de les diferents activitats per poder dur-les a terme i així concloure aquest treball satisfactòriament.

TAULA 3: QUADRE DE DATES

Fase 1	Durada	Inici	Final
Pre-Anàlisi	4 dies	22/09/16	26/09/16
Planificació	5 dies	27/09/16	02/10/16
Anàlisi	6 dies	03/10/16	09/10/16
Disseny i Prototip	12 dies	10/10/16	23/10/16
Fase 2	Durada	Inici	Final
Implementació	65 dies	17/10/16	15/01/17
Testeig unitari	12 dies	16/01/17	29/01/17

7 REQUERIMENTS

Com hem vist anteriorment, es van proposar una sèrie d'objectius específics pel projecte, donat que aquests eren generals, es van redactar una sèrie de requeriments pel projecte, que poguessin satisfer els objectius i necessitats, a més de donar una visió més específica d'aquests.

Tot seguit veurem els que es consideren més importants, tot i que es pot trobar un llistat més exhaustiu al document de requeriments creat en la primera fase d'anàlisi del problema.

7.1 El sistema ha de ser accessible des de dispositius mòbils

Aquest es un dels requeriments no funcionals més important donades les característiques del nostre projecte, on el

que es pretén és fer arribar les noves tecnologies, com poden ser els “smartphones”, a aquests tipus de jocs, de manera que es faciliti l'accés a la plataforma des de qualsevol navegador, per tal de poder accedir als personatges i fer modificacions, per exemple.

7.2 El sistema ha de ser implementant en Model Vista Controlador

Aquest requeriment no funcional, sorgeix d'una de les decisions presa a la fase de disseny i és que MVC [4], és un patró de disseny de software que divideix l'aplicació en tres blocs:

- Model: Controla la informació, lògica i regles de l'aplicació.
- Vista: Representa la informació que veurà l'usuari.
- Controlador: És l'intermediari entre el model i la vista.

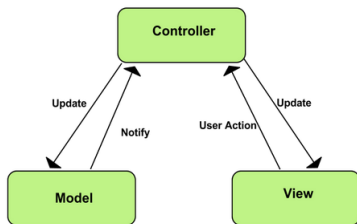


Fig 2: Interacció entre les parts de MVC

7.3 El sistema permetrà als usuaris crear, modificar i esborrar personatges de joc

La funcionalitat més bàsica del present treball, és oferir als usuaris la possibilitat de crear personatges per després administrar-los o afegir-los a sales privades, els personatges representen el mateix que a les partides en viu, així doncs es dona la possibilitat d'administrar els personatges d'aquestes partides des dels dispositius “smartphone”.

7.4 El sistema permetrà als usuaris crear sales privades

Donats els propòsits del treball, aquest requeriment funcional va ser un dels més importants en el moment del seu redactat, donat que es persegueix que els usuaris puguin representar en un entorn privat les interaccions existents en una partida de rol, d'aquesta manera ajudem als directores de jocs a portar un control sobre els personatges, per exemple, realitzar alguna acció sobre un personatge en concret sense que la resta sigui conscient.

8 ARQUITECTURA PLATAFORMA

En aquest punt explicarem l'arquitectura escollida pel desenvolupament de la plataforma, així com una visió global d'aquesta gràcies a aplicar la metodologia FDD i dissenyar un diagrama de classes “back-end” de la part del servidor.

Tenint en compte els requeriments i el fet de que la plataforma arribi als màxims dispositius possibles, l'arquitectura client-servidor va ser el model d'aplicació distribuïda que es va aplicar a l'hora de desenvolupar.



Fig 3: Arquitectura client-servidor

En aquest punt ja ens trobàvem aplicant la metodologia FDD, per tant abans de començar amb el desenvolupament, vam realitzar un diagrama de classes “back-end” que ens va donar una idea general del sistema, tenint en compte els objectius i requeriments del sistema.

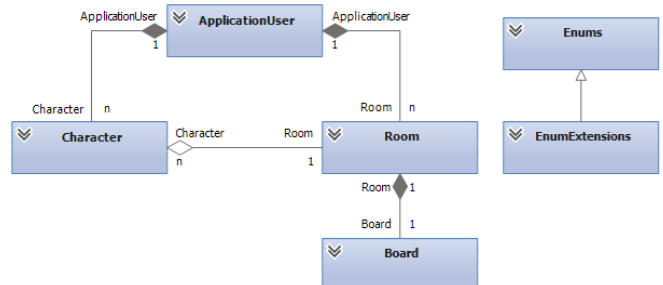


Fig 4: Diagrama inicial de classes “back-end” del servidor

9 EINES

En aquest punt exposarem algunes de les eines que es van utilitzar durant el desenvolupament del projecte i que han ajudat a la satisfacció del present treball.

9.1 Visual Studio

Visual Studio [5], va ser l'eina escollida pel desenvolupament del projecte, ja que incorpora unes plantilles que van permetre desenvolupar l'aplicació aplicant el patró MVC des de l'inici. A la figura següent podem apreciar que una solució inicial generada per l'eina separa els arxius en controladors, models i vistes.

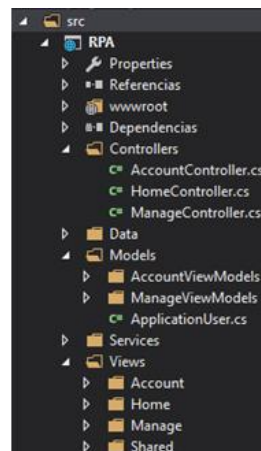


Fig 5: Solució inicial del projecte aplicant el patró MVC utilitzant VS

9.2 Team Foundation Server – TFS

TFS [6] ens facilita un repositori online, on podem fer els “check-in” del codi que anem desenvolupant, tenint així un control dels canvis i versions, a més d'una forma ràpida de recuperar el codi desenvolupat en el cas de la pèrdua local d'aquest. A més, aquesta eina s'integra fàcilment amb

Visual Studio de manera que ens facilita la manera de controlar el nostre codi. A la figura següent podem apreciar com es va integrar el repositori de TFS amb el Visual Studio, la plataforma de desenvolupament. A la part esquerra hi ha la ruta de repositori local, al centre la data de sincronització i a la dreta la ruta de repositori online.

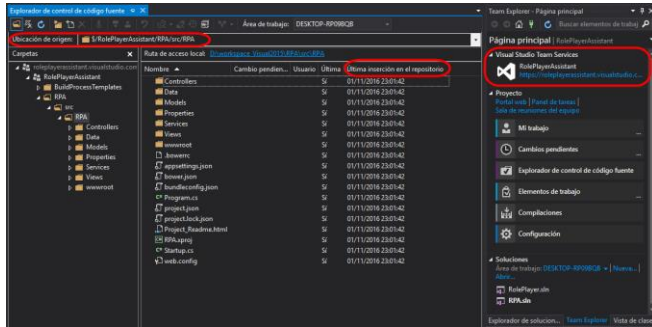


Fig 6: Finestra de Visual Studio de control de codi al repositori online de TFS

10 DESENVOLUPAMENT

Tenint en compte les classes anteriorment vistes, en aquest punt explicarem com s’han desenvolupat algunes característiques implementades a la plataforma, fent ús de la metodologia FDD. El fet d’aplicar aquesta metodologia suposa que per cada característica s’aplicarà una enginyeria del software tal i com explicarem més endavant.

10.1 Disseny i implementació dels usuaris

Amb la finalitat de donar a l’usuari la possibilitat de crear els personatges i sales desitjades, requerim d’aquest que es registri a l’aplicació informant el seu correu electrònic, una contrasenya i un pseudònim.

Per reutilitzar codi, es va aprofitar la API que incorpora Visual Studio quan es crea una solució amb les plantilles MVC, aquesta proporciona els mètodes necessaris per crear i controlar els usuaris, l’accés i sortida de l’aplicació, generador de codis per validar el correu electrònic o modificar la contrasenya.

A la imatge anterior, podem apreciar com l’aplicació valida els camps quan es vol registrar un nou usuari i indica els missatges d’errors pertinents, els quals estan prèviament definits al model de dades de l’entitat usuari, més endavant en aquest document podem veure alguns exemples d’aquestes configuracions de validacions.

Visual Studio també incorpora el seu SQL Server, el qual garanteix la integritat i seguretat de la base de dades, ja que s’incorpora al codi una inspecció de les dades que entren des de l’aplicació, comprovant que siguin verídiques i correctes, per evitar l’accés a usuaris malintencionats.

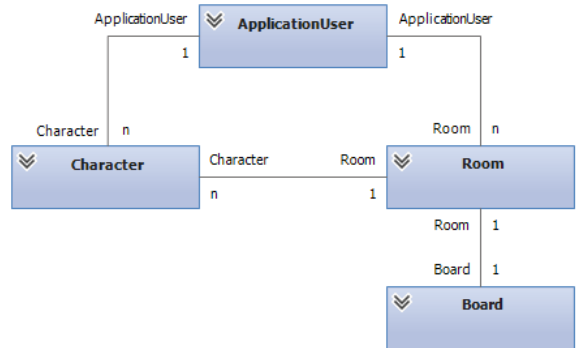


Fig 8: Diagrama E-R de la base de dades del servidor

A la figura anterior, podem veure el diagrama Entitat-Relació amb el que va ser dissenyada la base de dades del nostre servidor. A més de la clau primària i els atributs, cada entitat contindria una clau foranea amb l’identificador de l’entitat amb la qual està relacionada.

Per garantir correus electrònics autèntics, es requereix que els usuaris validin el seu correu electrònic. Visual Studio recomana l’ús de l’eina SendGrid [7], una aplicació de missatgeria que s’integra fàcilment amb el mètode generador de codis i a més de l’enviament de correus electrònics gratuïts, ens proporciona l’accés a una plataforma web, que disposa d’una sèrie de mètodes i informes per fer un seguiment dels correus electrònics enviats.

10.2 Disseny i implementació dels personatges

El que oferirem a l’usuari serà la possibilitat de crear personatges sense necessitat d’estar associats a una sala. Continuem aplicant la metodologia FDD i vam procedir a dissenyar per característica.

Primerament, abans de començar amb el desenvolupament de la característica, vam realitzar un digrama de casos d’ús de la característica a implementar tenint en compte els requeriments associats a aquesta, per poder visualitzar de manera global el seu comportament i preveure els mètodes necessaris per a la seva realització i posterior satisfacció dels usuaris que facin ús de la plataforma. A la figura següent podem veure aquest diagrama, en el que l’actor realitza les diferents accions disponibles a la característica.

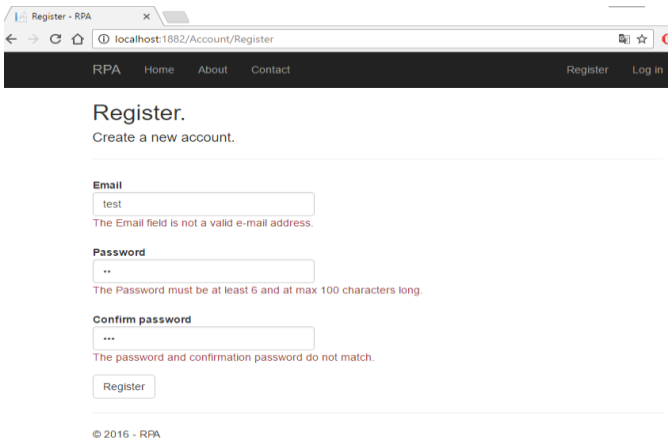


Fig 7: Vista de l’aplicació al registrar nou usuari informant els camps de manera errònia

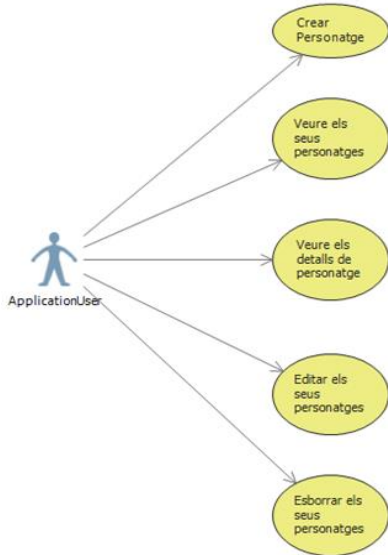


Fig 9: Diagrama casos d'ús de Personatge

Un cop realitzat el diagrama, es va passar a l'etapa de construir la característica; vam desenvolupar una classe model "Character.cs" que contindria els atributs de la classe, els quals, es configuren en el model per més tard aplicar les validacions necessàries quan s'emplenen formularis de creació o modificació. En la figura següent podem observar els atributs de la classe de personatges.

```

public class Character
{
    [ScaffoldColumn(false)]
    11 referencias
    public int CharacterId { get; set; }

    [Required]
    [Display(Name = "Tipo de Partida")]
    1 referencia
    public Enums.TipoPartida tPartida { get; set; }

    [Required]
    [Display(Name = "Reglas")]
    3 referencias
    public Enums.ReglasJuego Reglas { get; set; }

    [Required]
    [RegularExpression(@"^[a-zA-Z'\s]*$")]
    [StringLength(12, MinimumLength = 3)]
    [Display(Name = "Nombre")]
    1 referencia
    public string Nombre { get; set; }

    [Required]
    [Display(Name = "HP")]
    1 referencia
    public int HP { get; set; }

    8 referencias
    public virtual ApplicationUser User { get; set; }

    4 referencias
    public virtual Room Room { get; set; }
}
  
```

Fig 10: Model de dades de la entitat Personatge

Mitjançant aquest model, vam implementar el controlador amb els mètodes que actuarien d'intermediaris entre el model i les vistes, aplicant les restriccions necessàries a la lògica de l'aplicació per que cada usuari només pugui veure els seus personatges.

Per últim, es va desenvolupar les vistes per mostrar els personatges i les diferents accions que els usuaris poden

realitzar amb aquests.

Implementades totes les funcions necessàries de la característica, vam dissenyar i realitzar un joc de proves que comprovi la correcta funcionalitat del codi desenvolupat i que satisfaci les necessitats del projecte, podem veure una mostra a la figura següent.

ID prova	Req. ID	Nom	Descripció	Prerequisits	Accions	Resultat esperat	Estat prova
1	RF 7, 8	Crear nou personatge	Accedir al formulari de registre de personatge i crear un nou amb els camps obligatoris.	usuari: User existent a BD	Accedir a l'App mitjançant l'usuari dels prerequisits	S'obre l'App amb el nostre usuari	OK
					A la part superior de l'App fem click sobre "Characters".	Apareix la pantalla de visualització de personatges i observem els següents camps: - Create New (boto) - HP - Nombre - Reglas - Tipo de Partida	OK
					Fem click sobre "Create New"	Apareix la pantalla de creació de personatges amb els següents camps: - HP - Nombre - Reglas: D&D 3.5 (Per defecte) - Tipo de Partida: Plataforma (Per defecte)	OK
					Omplim els camps obligatoris i fem "click" al boto "save": - HP: 5 - Nombre: Test - Reglas: D&D 3.5 - Tipo de Partida: Plataforma	Tornem a la pantalla de visualització de personatges i apareix el que acabem de crear.	OK

Fig 11: Mostra de joc de proves de Personatge

10.3 Disseny i implementació de les sales

Acabada la implementació dels personatges, vam procedir a desenvolupar les sales, ja que oferim als usuaris la possibilitat de crear sales o unir personatges a sales existents. Si és el cas de crear una nova sala, l'usuari creador serà el director de joc d'aquesta. Si és un usuari que vol unir-se a una sala existent, es mostraran els personatges disponibles per introduir a aquesta, en cas que n'hi hagi.

A l'igual que vam fer amb la característica anterior, el primer pas va ser crear un diagrama de casos d'ús, que ens dones una idea general de com funcionaria la característica.

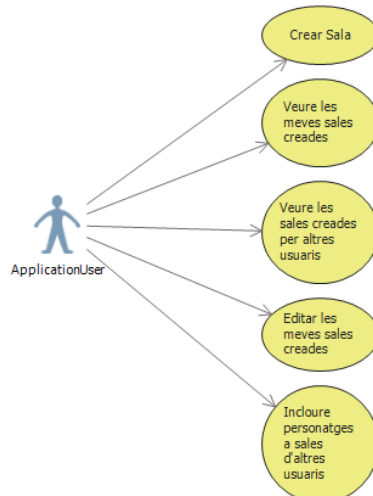


Fig 12: Diagrama casos d'ús de sales

Vam continuar amb el desenvolupament de la classe model "Room.cs", que contindria els atributs i configuracions necessaris, per realitzar les validacions oportunes en els formularis de creació i modificació. Com podem apreciar a la figura següent, aquesta classe incorpora un usuari i una col·lecció de personatges, que ens serviran per fer les distincions entre els tipus de jugadors de la sala.

```
public class Room
{
    [ScaffoldColumn(false)]
    13 referencias
    public int RoomId { get; set; }

    [Required]
    [Display(Name = "Tipo de Partida")]
    0 referencias
    public Enums.TipoPartida tPartida { get; set; }

    [Required]
    [Display(Name = "Reglas")]
    2 referencias
    public Enums.ReglasJuego Reglas { get; set; }

    [Required]
    [RegularExpression(@"^[a-zA-Z'\-\s]*$")]
    [StringLength(30, MinimumLength = 5)]
    [Display(Name = "Nombre de Partida")]
    0 referencias
    public string NombrePart { get; set; }

    [StringLength(120)]
    [Display(Name = "Descripción")]
    0 referencias
    public string Descripcion { get; set; }

    [Required]
    [Display(Name = "Número de jugadores")]
    2 referencias
    public int numPlayers { get; set; }

    [StringLength(120)]
    [Display(Name = "Reglas adicionales")]
    0 referencias
    public string descRules { get; set; }

    [Display(Name = "Master")]
    5 referencias
    public virtual ApplicationUser User { get; set; }

    6 referencias
    public virtual ICollection<Character> Characters { get; set; }
}
```

Fig 13: Model de dades de la entitat Sales

Per últim vam implementar el seu controlador i vistes, tenint en compte les restriccions que s’han d’aplicar a l’hora d’unir els personatges, ja que un usuari administrador d’una sala, no pot incloure els seus personatges en aquesta. Per altra banda s’han de mostrar els personatges adients a cada sala, i en cas que l’usuari no tingui personatges disponibles, donar-li l’opció de crear un de nou, sense la necessitat d’anar al menú específic de personatges. Per concloure amb la implementació de la nova característica, vam tornar a realitzar un joc de proves que la comproves.

ID prova	Req. ID	Nom	Descripció	Prerequisits	Accions	Resultat esperat	Estat prova	
1	RF 5, 13	Crear una nova sala correcte	Accedir al formulari de registre de sales i crear una nova amb els camps obligatoris.	usuari: User existent a BD	Accedir a l'App mitjançant l'usuari dels prerequisits	S'obre l'App amb el nostre usuari	OK	
					Apareix la pantalla de visualització de sales i observem els següents camps: - Create New (botó) - My Rooms As: Master / Player / All - Nombre de partida - Descripció - Reglas - Reglas adicionales - Núm. jugadores - Tipo de Partida - Master		OK	
					A la part superior de l'App fem click sobre "Rooms"	Apareix la pantalla de creació de sales amb els següents camps: - Nombre de partida - Descripció - Reglas: D&D 3.5 (Per defecte) - Reglas adicionales - Núm. jugadores - Tipo de Partida: Plataforma		OK
					Fem click sobre "Create New"	Omplim els camps obligatoris i fem "cli" al botó "create": - Nombre de partida: Test Partida - Descripció: Prova de crear - Reglas: D&D 3.5 - Reglas adicionales: No hi ha - Núm. Jugadores: 4	Tornem a la pantalla de visualització de sales i apareix la que acabem de crear, amb 2 nous botons "Edit"/"Delete".	

Fig 14: Mostra de joc de proves de Sales

A la figura següent podem apreciar l’arbre de la solució final del nostre projecte un cop implementades aquestes dues característiques.

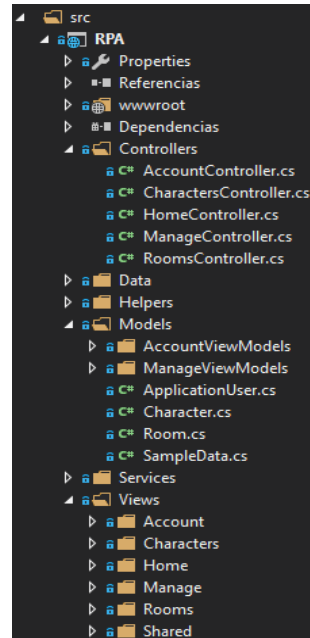


Fig 15: Solució final projecte MVC

10.4 Part comuna

Mentre es desenvolupaven les característiques vistes anteriorment va sorgir la necessitat d’implementar dues classes d’ajuda, amb la finalitat d’accedir més fàcilment a les diferents regles i tipus de partida, ja que s’utilitzen des dels controladors i les vistes.

En la figura següent podem apreciar la primera, que es tracta d’una classe amb els enumeradors que necessitem per diferenciar regles de joc o tipus de partida, en el cas de les regles de joc es van afegir configuracions per fer els seus enumeradors més “user friendly”.

```
public class Enums
{
    3 referencias
    public enum ReglasJuego {
        [Display(Name = "D&D 3.5")]
        [Description("D&D 3.5")]
        DD35,
        [Display(Name = "D&D 5.0")]
        [Description("D&D 5.0")]
        DD5,
        [Display(Name = "C'thulhu")]
        [Description("C'thulhu")]
        Cthulhu
    };
    3 referencias
    public enum TipoPartida { Plataforma, Presencial};
}
```

Fig 16: Classe d’ajuda de numeradors

Aquesta classe es va desenvolupar pensant en la possible extensió que es podria fer del present treball, i és que a l’igual que es dona la possibilitat a l’usuari d’escollir quines regles vol pel seu personatge o partida, es podria estendre aquesta funcionalitat aplicant les diferents regles de cada joc quan fos necessari, per exemple, aplicant modificadors a les característiques dels personatges exclusius de cada joc o introduir les diferents habilitats que pot realitzar cada personatge i de les que es disposa a cada joc.

En el cas de la segona classe, que veurem a la següent figura, es va desenvolupar per afegir als enumeradors el mètode que extrauria la informació inclosa en el camp de la descripció per mostrar-la a l'usuari.

```
public static class EnumExtensions
{
    Oreferencias
    public static string DisplayName(this Enum value)
    {
        if (value == null)
        {
            return null;
        }
        FieldInfo field = value.GetType().GetField(value.ToString());
        DescriptionAttribute[] attributes = (DescriptionAttribute[])field
            .GetCustomAttributes(typeof(DescriptionAttribute), false);
        if (attributes.Length > 0)
        {
            return attributes[0].Description;
        }
        return value.ToString();
    }
}
```

Fig 17: Mètode d'extensió per extracció de noms i descripcions

Amb aquest mètode vam poder mostrar els noms correctes a les vistes, ja que d'altra manera es mostraria el nom intern de l'aplicació a l'usuari.

11 TESTS DE VALIDACIÓ

Com hem pogut veure a la fase de desenvolupament, a mesura que anàvem introduint noves característiques a l'aplicació aquestes es testeaven fent ús de proves d'integració, però aquestes proves eren en entorns molt controlats. Per poder verificar alguns aspectes de l'aplicació de forma real, s'ha permet l'accés a la plataforma a un grup reduït d'usuaris.

Per verificar el compliment d'objectius s'han realitzat dos tipus de prova:

- Anàlisi de la interfície, compatibilitat amb diferents dispositius i usabilitat.
- Anàlisi de les característiques introduïdes.

Respecte a la interfície, compatibilitat i usabilitat, els comentaris rebuts han estat correctes i en la línia esperada, ja que la plataforma escollida per desenvolupar i el codi implementat, han estat orientats a oferir una interfície fàcil d'utilitzar i adaptable a diferents dispositius fent-la responsive.

En quant a les característiques introduïdes, han complert amb les necessitats dels usuaris a l'hora d'utilitzar una plataforma com aquesta, els comentaris per parts dels usuaris retornant que les característiques funcionen correctament i amb un bon temps de resposta.



12 RESULTATS

En aquest punt exposarem els resultats obtinguts a l'acabar el present projecte, donant una explicació a aquells punts no complerts o parcialment realitzats.

Com hem vist al principi es van redactar una sèrie d'objectius per la satisfacció del projecte, tot i que hem complert

la majoria d'aquest, a la taula següent podem veure els que no hem assolit totalment.

TAULA 3: VALORACIÓ D'OBJECTIUS

Objectius	Compliment
Donar autonomia a les partides sense necessitat de director de joc.	
Satisfer els requeriments especificats al document de requeriments.	






12.1 Donar autonomia a les partides

Es va plantejar com a objectiu secundari i finalment no ha sigut possible la seva satisfacció. Això ha sigut pel temps disposat per la realització del projecte, donat que al final s'ha hagut de dedicar més temps del previst a alguns aspectes del desenvolupament de les característiques mencionades prèviament i el fet de l'aleatorietat que suposa una partida de rol, els jugadors poden fer accions il·limitades i s'hauria de programar una IA capaç de donar resposta, com a mínim, per les accions més habituals dins d'una partida.

12.2 Satisfer els requeriments especificats

Tot i que hem satisfet la gran majoria de requeriments especificats al seu document, existeixen alguns no satisfets que veurem a taula següent.

TAULA 4: VALORACIÓ REQUERIMENTS

Requeriments no funcionals	Compliment
El sistema ha de ser implementat multiidioma.	
Requeriments funcionals	Compliment
El sistema permetrà al usuaris comunicar-se entre ells públicament mitjançant un fòrum.	
El sistema permetrà als usuaris comunicar-se entre ells privadament amb missatges de text.	
El sistema permetrà als usuaris comunicar-se en les sales privades mitjançant un chat.	
El sistema permetrà el llançament de daus dins les sales privades mitjançant un algoritme aleatori de daus.	

12.2.1 Sistema multiidioma

Tot i que el codi està preparat per suportar aquesta funcionalitat, per falta de previsió final, no s'han pogut realitzar els arxius que donen suport a altres idiomes, fent així que aquest requeriment no estigui de forma inicial a l'aplicació.

12.2.2 Comunicació via fòrum

En aquest cas quan es va realitzar l'anàlisi inicial, es va tro-

bar el plugin OpenForum [9], un fòrum fàcilment adaptable a projectes MVC de Visual Studio, el problema es va donar a l'hora d'integrar el plugin al nostre projecte, ja que existia una incompatibilitat amb la versió de .NET entre el plugin i el projecte. Es va procedir a realitzar un anàlisi més exhaustiu i es va trobar el plugin POPForums [10], amb el mateix resultat final, error d'incompatibilitat.

La solució més viable sense refer el projecte és desenvolupar el nostre propi fòrum per l'aplicació, ja que no podem saber si al començar el projecte de zero, utilitzant la versió de .NET compatible amb qualsevol dels dos plugins mencionats, podríem reutilitzar el codi desenvolupat fins al moment. Però per poder aplicar aquesta solució, hauríem de deixar de banda altres funcionalitats més importants pel projecte.

12.2.3 Comunicació via missatge privat

Donat l'extra de temps dedicat a la tasca anterior, no contemplat a priori, es va sacrificar aquesta funcionalitat, ja que es va considerar més interessant el fet que, els usuaris poguessin compartir les seves opinions i històries públicament.

12.2.4 Comunicació sales privades

Els usuaris tenen la possibilitat de comunicar-se, però el funcionament no és exactament el d'un "chat", ja que per veure si altres usuaris han escrit algun nou missatge, hem de refrescar la pàgina. Per evitar aquest detall es volia utilitzar la llibreria de ASP.NET SignalR [11], la qual estableix comunicació en temps real entre servidor i client, a més incorpora una API per administrar les connexions, connectar i desconnectar o realitzar agrupacions de connexions.

A l'igual que amb la comunicació via fòrum, el problema va aparèixer a l'integrar la llibreria al nostre projecte, i és que quan es va crear el projecte, la versió de .NET que es va seleccionar era massa actual, per tant aquesta llibreria encara no disposa de la compatibilitat amb la versió escollida.

La solució que es va trobar, va ser aplicar Web Sockets per crear una comunicació en temps real, però donada la limitació de temps del projecte no va ser possible realitzar-la a temps.

12.2.5 Llançament de daus

D'igual forma que passa amb comunicació a sales privades, tenim desenvolupat el sistema de llançament de daus, però per poder visualitzar qualsevol tirada realitzada per un altre usuari, ens veiem forçats a refrescar la pàgina, seria desitjable que fos una comunicació en temps real.

Es podria solucionar fàcilment amb el mateix mètode de Web Sockets comentat anteriorment.

13 CONCLUSIONS FINALS

Per acabar el treball presentarem les conclusions del procés efectuat al llarg d'aquests cinc mesos. Per últim indicarem possibles continuacions del treball efectuat.

Com a conclusió global, podem extreure que gràcies a aplicar enginyeries del software sobre el treball que hem hagut de realitzar, hem aconseguit portar d'inici a fi el projecte de software que hem exposat en aquest article, i que a pesar de les dificultats trobades, gràcies a una bona planificació i l'aplicació de certes metodologies hem aconseguit uns bons resultats en la finalització del treball.

13.1 Valoració dels objectius

Amb el desenvolupament de les característiques implementades corresponents als usuaris, personatges i sales hem pogut oferir el tipus de plataforma web que desitjàvem, doncs les persones que l'utilitzin poden guardar els seus personatges i partides, i accedir a ells des de qualsevol dispositiu apropant així les noves tecnologies a aquest tipus d'activitat i incorporant ajudes que d'altre manera no seria possible.

13.2 Valoració dels requeriments

Hem de considerar que tot i els problemes trobats durant el desenvolupament del projecte, hem realitzat una bona feina, ja que hem satisfet gran part dels requeriments especificats al seu document i gràcies això trobem els bons resultats presentats a la fase de testeig unitari, que vénen donats en part per la metodologia utilitzada durant el desenvolupament, ja que ens proporcionava la implementació d'unes característiques robustes i ben testejades.

14 TREBALL FUTUR

Per tal de donar una continuïtat al present treball, podríem proposar una sèrie de possibles extensions, que poden ser útils per millorar l'experiència dels usuaris a la nostra aplicació.

Primerament s'hauria de realitzar un estudi per comprovar quina solució és més adient per resoldre els problemes de compatibilitat del fòrum i el SignalR, ja que fer el codi de zero del fòrum i aplicar Web Sockets al projecte actual, pot resultar una tasca més costosa que el fet de reiniciar el projecte utilitzant la versió de .NET adient per les dues funcionalitats i reutilitzar tot el codi desenvolupat fins ara.

Pel que fa al sistema multiidioma, s'hauria de dedicar temps a crear els arxius de traduccions dels idiomes desitjats i activar a les vistes el reconeixement d'idioma que els hi proporciona l'explorador web de l'usuari connectat.

Continuant amb la línia de desenvolupament del projecte, es podria estendre la funcionalitat de regles, per no només separar els personatges i les sales, sinó també per aplicar els diferents tipus de regles a les partides que es duguin a terme a les sales, així s'hauria d'incloure al sis-

tema tota la informació relacionada amb les regles importants dels diferents sistemes de joc, per més tard poder aplicar-les a les sales privades, taulers i personatges, de manera que cadascun tingui més informació rellevant pels usuaris.

Per altra banda, es podria implementar una nova funcionalitat en els taulers, que envies una alerta a l'usuari al qual li toca realitzar acció, ja sigui mitjançant un correu electrònic o una alerta a la plataforma.

Crear una IA donaria una nova funcionalitat, la qual cap altre pàgina similar disposa, de narració de partides sense necessitat d'un director de joc, d'aquesta manera la IA seria l'encarregada d'anar presentant la història als jugadors i adaptar-se a les possibles accions que aquest duguin a terme.

Una altre possible extensió seria l'aplicació de campanyes de màrqueting per anunciar noves funcionalitats per exemple, gràcies a l'eina de missatgeria presentada, SendGrid, la qual ens ofereix un alt seguiment dels correus electrònics que enviem i podem configurar llistes pel posterior enviament de determinats missatges.

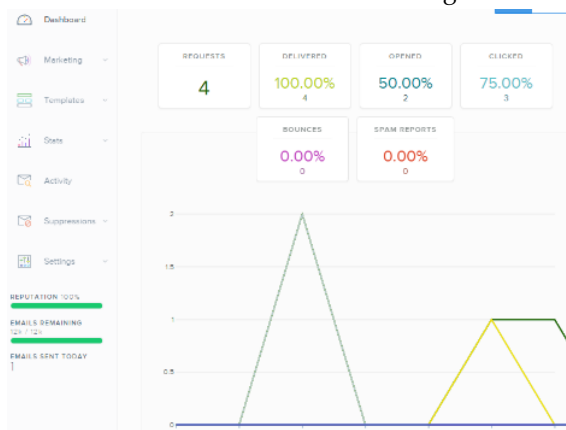


Fig 18: Vista general de SendGrid

AGRAÏMENTS

Agrair a Lluís Gesa Bote pels bons consells i suport donats durant el transcurs del present treball i que han fet possible la realització d'aquest. Agrair a la meua família el seu suport incondicional durant la meua etapa com a universitari, i a la meua novia per ajudar-me i donar-me suport en els moments més difícils.

Finalment, m'agradaria donar les gràcies a companys i amics pels bons moments i facilitats que m'han atorgat durant aquesta etapa.

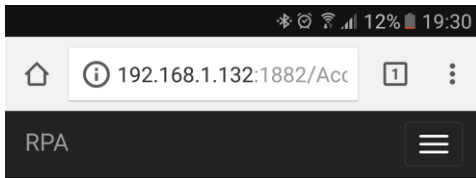
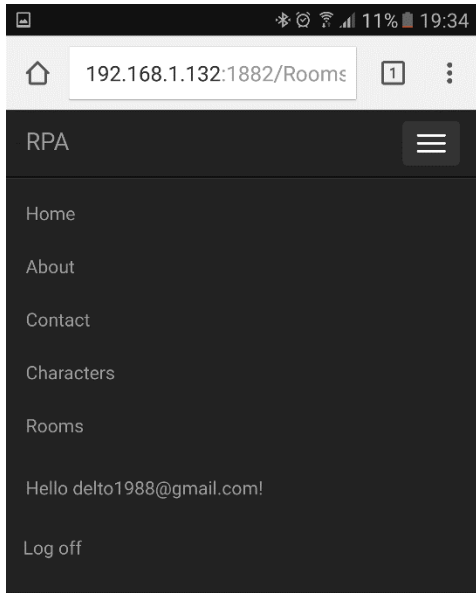
BIBLIOGRAFIA

- [1] Comunidad Umbria. Enllaç: <http://www.comunidadumbria.com/> [Consulta: 21/09/2016]
- [2] MUD Medina. Enllaç: <http://www.angelfire.com/mb/medina/general.html> [Consulta: 21/09/2016]
- [3] Feature-driven development (FDD) and Agile Modeling. Enllaç: <http://www.agilemodeling.com/essays/fdd.htm> [Consulta: 25/09/2016]

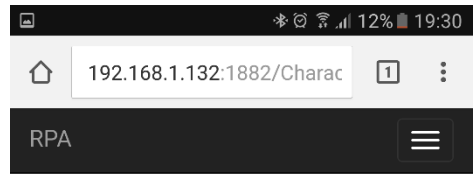
- [4] Chrome developer MVC Architecture. Enllaç: https://developer.chrome.com/apps/app_frameworks [Consulta: 03/11/2016]
- [5] Microsoft ASP.NET. Guia MVC per Visual Studio. Enllaç: <https://docs.asp.net/en/latest/mvc/index.html> [Consulta: 10/10/2016]
- [6] Microsoft Team Foundation Server. Enllaç: <https://www.visualstudio.com/tfs/> [Consulta: 26/10/2016]
- [7] SendGrid. Enllaç: <https://sendgrid.com/> [Consulta: 10/10/2016]
- [8] ASP.NET MVC 5 Internationalization. Enllaç: <http://afana.me/post/aspnet-mvc-internationalization.aspx> [Consulta: 10/10/2016]
- [9] OpenForum – MVC Forum. Enllaç: <http://openforum.codeplex.com/> [Consulta: 10/10/2016]
- [10] POPForums. Enllaç: <https://github.com/POPWorldMedia/POPForums/wiki/Documentation> [Consulta: 10/12/2016]
- [11] ASP.NET SignalR. Enllaç: <https://www.asp.net/signalr> [Consulta: 09/10/2016]

APÈNDIX

A1. INTERFICIE D'USUARI VISTA DES D'UN DISPOSITIU MÒBIL



© 2016 - RPA

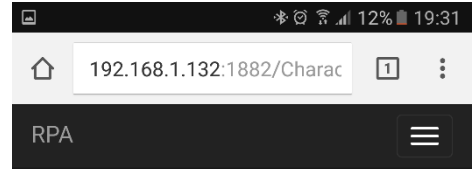


Index

[Create New](#)

HP	Nombre	Reglas	Tipo de Partida	
5	Test	D&D 3.5	Plataforma	Edit Details Delete
123	asd	D&D 5.0	Presencial	Edit Details Delete

© 2016 - RPA



Create

Character

HP

Nombre

Reglas

Tipo de Partida

[Create](#)

[Back to List](#)

Index

Create New

My Rooms AS: Master Player All

Nombre de Partida	Descripción	Reglas	Reglas adicionales	Número de jugadores	Tipo de Partida	Master
Test master	prueba que el master se setea	D&D 3.5	test	7	Plataforma	delto1988@gmail.com <a>Edit <a>Delete
Test para unir player	que se recupera de bd	D&D 3.5		2	Plataforma	delto_88@hotmail.com <a>Join <a>Details
test unir player dd	dd5	D&D 5.0		3	Plataforma	delto_88@hotmail.com <a>Join <a>Details
Test Partida	Prova de editar	D&D 3.5	no hi han	4	Plataforma	delto1988@gmail.com <a>Edit <a>Delete

© 2016 - RPA

RPA

Create Room

Nombre de Partida

Partida test

Descripción

Test

Reglas

D&D 3.5

Reglas adicionales

Número de jugadores

5

Tipo de Partida

Presencial

Create

2

Tipo de Partida

Plataforma

Characters in

HP	Nombre	Reglas	Tipo de Partida
5	Test	D&D 3.5	Plataforma

Remove | Details

Select your character

HP	Nombre	Reglas	Tipo de Partida
0	TEST	D&D 3.5	Plataforma

Join | Details

Back to List