



UNIVERSITAT DE
BARCELONA

Treball final de grau

GRAU D'INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Anàlisi de l'aplicació de la combinació de metodologies àgils

Autor: Xavier Rodríguez Montesinos

Director: Dr. Eloi Puertas i Prats

Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 13 de setembre de 2020

Abstract

This project presents a new software development methodology.

This methodology, based in another existing two (Scrum and Pomodoro), tries not only to improve efficiency and performance on teams, but to also improve the individual worktime of each of the members that belongs to that teams. That is why, with this proposal, we want to achieve goals with less time than the needed with the current methodologies.

To do so, we have created a tool (web app) that will be used to control how the developers are being adapted to the methodology, how they progress with their tasks while they are using it and the time they need to finish them.

With this tool, we have elaborated an experiment with some members of a real development team in order to check its performance.

After finishing the experiment, we have observed that this idea can result very positive in a controlled enviroment which is able to implement accurately its rules. But, if this environment where we try to implement the methodology, does not follow its principles, we will not get these positive results compared from using other methodologies like Scrum or Kanban.

Resum

Aquest treball planteja una nova metodologia en l'àmbit del desenvolupament de software.

Aquesta metodologia, derivada de dues altres ja existents (Scrum i Pomodoro), intenta no només millorar l'eficiència i rendiment en aspectes d'equip, sinó també en el temps individual de cadascun dels membres que el formen. Per tant, amb aquesta proposta, podem arribar a obtenir objectius amb un temps molt menor al que trobem en moltes metodologies àgils.

Per tal de provar aquesta hipòtesi, hem elaborat una eina (app web) amb la qual poder controlar com els desenvolupadors s'adapten a la metodologia, com progressen amb les seves tasques mentre la utilitzen i els temps de finalització d'aquestes.

Amb aquesta eina, hem realitzat un experiment amb diversos membres d'un equip real de desenvolupament de software per tal de comprovar-ne la seva eficàcia.

Després de finalitzar l'experiment, hem vist que aquesta idea pot obtenir resultats molt positius en àmbits controlats i que s'ajustin molt bé a ella. En canvi, si l'ambient on s'inclou aquesta metodologia no es regeix per les regles que estableix, els resultats no seran més positius respecte l'ús altres metodologies com poden ser Scrum o Kanban¹.

¹Kanban: metodologia de treball que consisteix en un seguit d'estats a través dels quals es mouen les tasques a desenvolupar. Per exemple: "Per fer", "En progrés" "Acabat".

Agraïments

Vull agrair a tot l'equip involucrat en l'experiment del projecte pel seu esforç i dedicació durant unes 4 setmanes d'aprenentatge i noves experiències. Espero que aquesta no sigui l'última vegada que ens endinsem en territoris inexplorats i que junts puguem millorar dia rere dia.

També és important l'especial menció d'aquelles persones i acompanyants que han fet més lleugera tota aquesta càrrega.

Índex

1	Introducció	1
2	Objectius	3
2.1	Objectius del projecte	3
2.2	Objectius de la metodologia	4
3	Planificació	5
4	Anàlisi	6
4.1	Scrum - La metodologia estrella	6
4.2	La tècnica Pomodoro - Descobreix què és el temps	10
5	Disseny	16
5.1	La nova metodologia	16
5.1.1	Un cicle de la nova metodologia	17
5.2	L'eina	22
5.2.1	Requisits	22
5.2.2	Model de dades	23
5.2.3	La vista	27
6	Implementació	39
6.1	Entorn de desenvolupament	39
6.2	L'aplicació	40
6.2.1	El temporitzador, nucli del Pomodoro	40
6.2.2	Models i usuaris	42
6.2.3	Vista	42
7	Experiment	43
7.1	Plantejament	43
7.2	Persones	43
7.3	Entorn i eines	44
7.4	Entrenament	44
7.5	Primera iteració	44
7.6	Segona iteració	45

7.7	Tercera iteració	46
7.8	Quarta iteració	46
7.9	Resultats	47
7.10	Conclusions i treball futur	51
A	Annex	53
A.1	Model de dades	53
A.2	Consentiment informat	54
A.3	Informes diaris de treball	55

1 Introducció

Context

Durant la dècada dels anys 60, van sorgir les primeres metodologies de desenvolupament. Aquestes, tenien l'objectiu de millorar els temps de finalització del producte per tal d'abaratir els costos. Per tal de fer això, van haver de crear un seguit de normes i procediments molt estructurats, amb el fi de tenir control sobre tot el que estava passant en cada moment en l'empresa.

Des de llavors, han sorgit un gran nombre de metodologies. D'aquestes, moltes han acabat en desús per mancances o incompatibilitats. Però d'altres han acabat sent el dia a dia de molts desenvolupadors i treballadors en general, com per exemple la metodologia en cascada (Waterfall²) o la metodologia Scrum (molt ben contemplada actualment).

Avui en dia, i en el món del desenvolupament de software, després que totes aquestes metodologies hagin estat establertes (fins i tot algunes com a estàndard), s'ha pogut millorar molt el rendiment dels equips a l'hora de complir objectius i dates. I no només això, sinó que també s'ha arribat a poder estimar amb prou certesa quant un equip pot trigar en acabar un projecte o una funcionalitat.

Tot i això, un problema que ens seguim trobant encara és la poca optimització del temps de l'individu, del treballador. Una persona, durant la seva jornada laboral, pot arribar a perdre una gran quantitat de temps degut a les contínues interrupcions que succeeixen al seu voltant. Aquests fets normalment porten a una baixa productivitat del treballador.

Motivació

La nostra proposta és adaptar, en aquestes metodologies que es fan servir actualment, altres tècniques de millora de productivitat individual.

Hem escollit una metodologia i una tècnica concretes per tal de fer més fàcil i visible al lector veure com funcionen per separat i en conjunt. Tot i que explicarem ambdues parts en detall de forma separada i un cop juntes, hi ha molta informació arreu que es pot consultar en cas de necessitat.

Primer, la metodologia de desenvolupament que hem escollit és la coneguda Scrum, que és una proposta sòlida, estructurada i ben repartida entre els membres d'un equip de desenvolupament.

D'altra banda, la tècnica d'optimització del temps escollida ha sigut la tècnica Pomodoro. Aquesta, basada en el timeboxing³, permet millorar l'eficiència d'un treballador mitjançant la concentració individual en una tasca.

²Waterfall: metodologia de gestió de projectes lineal. S'anomena *cascada* perquè les diferents etapes de desenvolupament depenen de les anterior i, per tant, cauen en cascada.

³Timeboxing: tècnica de gestió del temps que consisteix en definir una unitat màxima de temps, i aleshores desenvolupar una tasca en aquest temps establert.

Estat de l'art

Per separat, la metodologia i la tècnica són fortament conegudes. Moltes empreses les han implementat i han obtingut resultats molt positius.

La metodologia Scrum es porta utilitzant des de finals dels anys 80, pel que la informació sobre aquesta és accessible i la podem trobar en grans quantitats.

El mateix podem dir sobre la tècnica Pomodoro, inventada per Francesco Cirillo, que ha estat utilitzada no només per empreses i en àmbits de desenvolupament, sinó també en el dia a dia d'estudiants i treballadors de tot tipus.

Després de la recerca d'una hibridació de les dues, no n'hem trobat gaire informació al respecte. Per tant podríem dir que o bé ningú havia pensat abans en aquesta possibilitat, o bé després de provar-la no n'han sortit gaire satisfets.

El que sí vam trobar és un article[2] que parla sobre la implementació de la tècnica Pomodoro en metodologies àgils per tal de reduir les interrupcions dels desenvolupadors. Tot i ser curt (4 pàgines), ens mostra una idea de com funciona la tècnica i una forma d'adaptar-la a aquestes metodologies.

Aquest comença explicant l'ús de les metodologies de gestió del temps i com ajuden als individus a millorar la manera d'aprofitar el seu temps. A continuació en fa un breu resum sobre els seus principis (blocs fixos de temps que no es poden interrompre) i el seu funcionament (25 minuts de treball i 5 o 15 de descans). També esmenta la importància d'evitar les interrupcions i un seguit d'eines i activitats a realitzar per tal d'utilitzar la tècnica.

Durant l'article podem observar diversos exemples i taules on se'ns explica com aquests procediments han de dur-se a terme, així com algunes gràfiques amb resultats que ens ensenyen, per exemple, com l'ús d'aquesta tècnica fa que decreixin amb el temps les interrupcions en els treballadors d'un equip.

Finalment, conclou amb que la tècnica Pomodoro és cada cop més usada en entorns de desenvolupament, i que han observat, com comentàvem, una disminució de les interrupcions, però també el temps necessitat per a cada cicle de desenvolupament, cosa que provocava un augment de la satisfacció dels treballadors, de la mateixa manera que ho feia la dels responsables d'aquests. Un problema que hi veuen, però, és una difícil comunicació entre els membres de l'equip, que a vegades pot provocar bloquejos en les seves tasques.

Aquests serien alguns dels resultats que esperem obtenir també durant la realització de l'experiment.

2 Objectius

Podríem dividir els objectius entre els propis del projecte i els de la nova metodologia.

2.1 Objectius del projecte

Detallem a continuació els objectius del projecte i com hem pensat abordar-los.

- Comprendre la metodologia que utilitzarem com a base per a definir la nostra nova solució híbrida - Scrum.
 1. Investigar les seves bases, la seva estructura i el seu funcionament.
- Comprendre la tècnica d'optimització de temps amb la qual combinarem l'anterior metodologia - La Tècnica Pomodoro.
 1. L'autor de la tècnica - Francesco Cirillo - ens porta a través del seu llibre[1] a tots els racons de la seva tècnica.
- Definir una nova metodologia resultant de la combinació de les dues anteriors.
 1. Extreure de ambdues metodologies les parts comuns que es podrien barrejar.
 2. Adaptar el cicle del desenvolupament a aquesta hibridació.
 3. Definir un seguit de normes que representin aquesta nova metodologia.
- Elaborar una eina que, seguint els principis de la nostra nova metodologia, ens permeti quantificar el rendiment que se li pot treure i, per tant, poder comparar-la amb altres metodologies.
 1. Creació d'una aplicació web que els usuaris podran utilitzar amb la finalitat d'aplicar la nova metodologia.
- Provar l'eina produïda en un entorn real per tal d'obtenir resultats quantitatius.
 1. Definició d'un experiment.
 2. Realitzar l'experiment i guardar-ne dades dins l'aplicació implementada.
- Analitzar els resultats de l'experiment realitzat i treure conclusions sobre la nova metodologia.
- Comparar la metodologia base (Scrum) amb la hibridació en termes de rendiment i resultats.

2.2 Objectius de la metodologia

- Obtenir resultats ràpidament.
 1. Seguint els principis de *Scrum* i les metodologies àgils.
- Disminuir les distraccions dels treballadors.
 1. Seguint els principis de la tècnica *Pomodoro*.
- Incrementar el màxim possible el temps treballat i, per tant, el rendiment dels treballadors.
 1. La unió entre les dues parts ens donarà rapidesa a l'hora d'obtenir resultats gràcies a que ajuntarem una metodologia àgil amb una tècnica de optimització del temps individual.

3 Planificació

La durada del projecte s'ha vist allargada per esdeveniments externs a ell que han succeït durant la seva realització, com ha sigut la pandèmia del COVID-19. Tot i això, gràcies a la naturalesa d'aquest, no ens hem vist obligats a aturar-lo. Fins i tot, hem pogut polir i dedicar més temps a certes parts que la requerien.

Aquestes serien les dates orientatives que van estar planificades per la finalització del projecte:

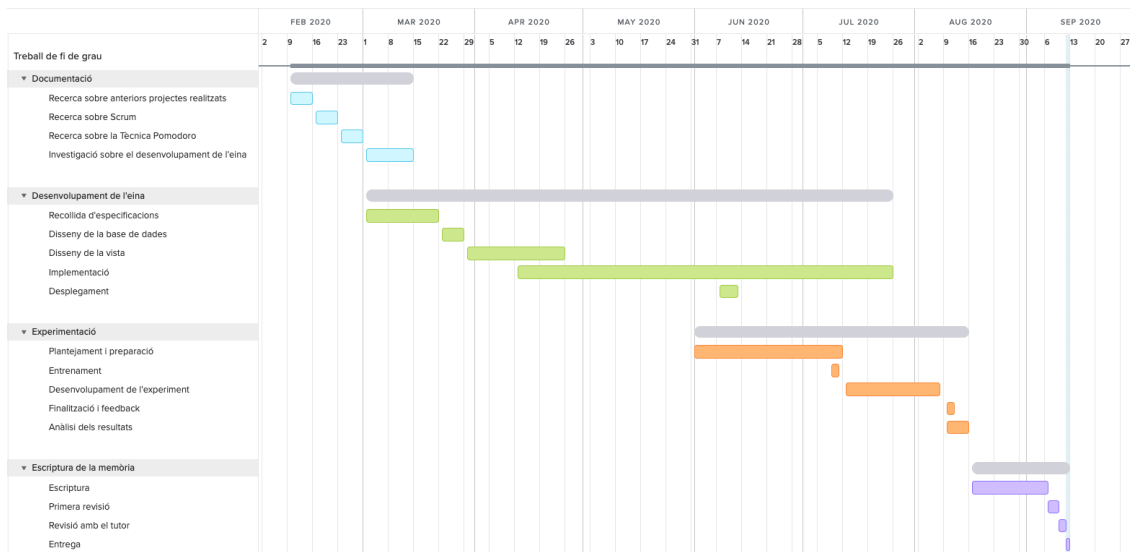


Figura 1: Planificació del projecte

Hem dividit les tasques en 4 apartats:

- Documentació: tasques relacionades amb recaptació d'informació prèvia al desenvolupament de la metodologia i l'eina per provar-la.
- Desenvolupament de l'eina: disseny i desenvolupament de l'aplicació que ens permetrà utilitzar la metodologia.
- Experimentació i anàlisi de resultats: fase on realitzarem l'experiment. Per tal de començar-la necessitarem les dues anterior parts finalitzades.
- Escriptura de la memòria: redacció, revisió i entrega de la memòria del projecte.

Algunes d'aquestes tasques tindran una durada curta (1 setmana aprox.), ja que impliquen feines senzilles de recerca. Altres, com per exemple el disseny i desenvolupament de l'aplicació o la realització de l'experiment, tindran una durada considerablement llarga degut a l'alta càrrega de feina que impliquen o, en el cas de l'experiment, perquè tenen una llargada definida.

Per a la redacció de la memòria hem guardat les últimes 4 setmanes del termini, tal i com es recomanava.

4 Anàlisi

4.1 Scrum - La metodologia estrella

Scrum és una manera de desenvolupar, d'entregar un producte donades unes especificacions.

Hem dit diversos cops que Scrum és una metodologia, però, segons els seus creadors Ken Schwaber i Jeff Sutherland, no ho és. Podem definir-lo com un *framework*⁴ que la gent o un equip de persones pot utilitzar per resoldre problemes complexos.

Està categoritzat dins de les metodologies àgils⁵, les quals es basen en el desenvolupament iteratiu o incremental. Els requeriments i especificacions van evolucionant amb el temps, per això és intel·ligent seguir un cicle que comenci amb la recollida d'aquests i acabi amb l'anàlisi dels resultats, per així veure si cal tornar a repensar una cosa o canviar alguna part del que s'ha implementat. Però en aquest apartat ens volem centrar exclusivament en l'Scrum, així que deixarem a la decisió del lector si endinsar-se més o no en les bases d'aquest.

El seu èxit es basa en la facilitat d'adaptar-se als diferents entorns que podem trobar en un equip de desenvolupament, a les seves regles i els seus rols ben definits. Sí que és veritat que si analitzem les empreses, organitzacions o equips que l'utilitzen, trobarem diferències a l'hora d'aplicar-lo, però les bases i la funcionalitat principal acaben sent les mateixes.

Per a que tot això segueixi el seu transcurs, calen un seguit de components dins de l'equip que s'encarreguin de diverses tasques. A vegades, aquests poden adoptar més d'un rol si és necessari, però en projectes complexos el millor és que cada membre s'encarregui exclusivament de la seva part, ja que quan la càrrega laboral d'una persona excedeix dels seus límits, les conseqüències normalment recauen en tot l'equip i, per tant, en el producte que estan desenvolupant.

Un equip de desenvolupament pot tenir un gran nombre de persones implicades, de diferents categories i rangs. Però imaginem que tenim un equip més reduït, en el que no inclourem ni directius, ni personal no estrictament relacionat amb el desenvolupament del producte. Dins d'Scrum veurem en aquest equip una divisió clara en els rols que cadascú interpreta.

En primer lloc necessitem algú que doni les especificacions del producte, com ha de ser (quina forma, quin color, el flux d'una funcionalitat, etc.), a aquest l'anomenarem *Product Owner* - el propietari del producte - que no necessàriament és qui posa els diners perquè aquest es dugui a terme, sinó que és l'encarregat que el resultat sigui l'esperat.

Seguidament, ja que utilitzem Scrum, una persona que s'encarregui de que tot-

⁴Framework: infraestructura composta per diversos elements preexistents que facilita la realització d'un conjunt de tasques.

⁵Metodologies àgils: conjunt de metodologies de desenvolupament de software basades en el desenvolupament iteratiu, on els requeriments i les solucions evolucionen gràcies a la col·laboració entre diversos equips.

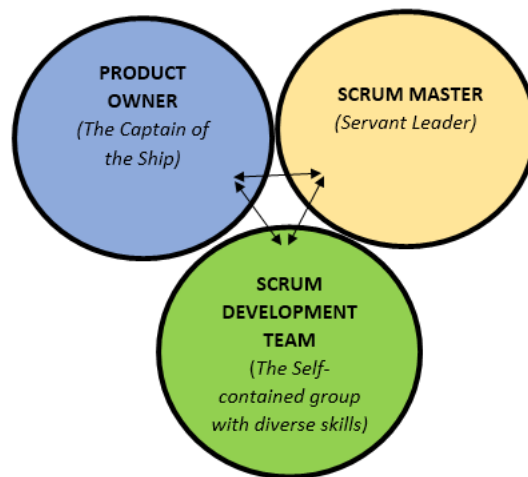


Figura 2: Principals rols dins del Scrum

hom segueixi els seus principis i regles, i veure si algú té un impediment en les seves tasques (també anomenades *històries d'usuari*) o en el seu equip. Per tant, aquesta persona ha de tenir un gran coneixement del *framework* per poder, no només aplicar-lo ell mateix, sinó que tothom en l'equip ho faci també. Per això, se l'anomena *Scrum Master*.

I, finalment, el *Scrum Team*. Una persona o conjunt de persones que seran els que s'encarreguin de realitzar aquelles tasques que, gràcies als dos anteriors, s'han pogut definir i repartir adequadament. Aquest conjunt, si està format per diverses persones, pot contenir una gran diversitat de coneixements realment útils per el desenvolupament del producte. Per això, és important que l'encarregat de repartir les tasques conegui el nivell i habilitats de cadascun d'ells.

Com hem dit anteriorment, l'Scrum és dins de les metodologies àgils, i com a tal segueix un desenvolupament iteratiu.

Com podem veure en la imatge anterior, el cicle del Scrum està dividit en diverses parts, però, a menys que s'acabi definitivament el producte, la fletxa ens torna un altre cop a un dels estadis inicials, seguint així la contínua iteració de les metodologies àgils. Cada iteració l'anomenarem *Sprint*.

En cada part del cicle es treballa una petita part del desenvolupament, i això fa possible la bona organització del projecte.

Quan s'inicia un nou projecte, el que normalment es fa és obtenir els requeriments del producte final a través del *Product Owner* i l'equip directiu adient. Un cop obtinguts aquests requeriments, aquests es transformen en tasques que es llisten en una llista anomenada *Backlog*. Aquesta va creixent a mesura que s'avança en el desenvolupament, sorgeixen propostes noves o hi ha algun canvi en els requeriments inicials.

Després d'això, cal reunir l'equip i preparar el primer cicle de la iteració en el que anomenem *Sprint Planning*. La durada d'aquests cicles no està definida enlloc, però normalment s'opta per una setmana o dues. Aquesta reunió, de llargada



Figura 3: Cicle del Scrum

considerable (4 hores aproximadament, tot i que a la pràctica en són menys), està formada com a mínim per l'*Scrum Master* i pels membres de l'*Scrum Team*, i consisteix en agafar les tasques amb més prioritat, avaluar la seva complexitat, i decidir si entraran o no en aquest cicle. Per tal d'avaluar la complexitat de les tasques, es poden utilitzar diferents mètodes que normalment es basen en l'estimació i el consens entre els membres de l'equip. Amb el temps, tècniques com el *Poker planning*⁶ han sorgit, facilitant i fent més entretinguda aquesta feina. Al final, s'haurà elaborat un subgrup del *Backlog* anomenat *Sprint Backlog*. Tot i que no és estrictament necessari, molts equips ja aprofiten aquesta reunió per assignar les tasques escollides als diferents membres de l'equip.

Un cop feta aquesta planificació, es pot començar amb el cicle o *Sprint*. Durant aquest és quan els membres de l'equip treballen per acabar com abans millor les seves tasques. Per tal d'ajudar-los, diàriament es fa una curta reunió (màxim 15 minuts), anomenada *Daily Scrum* o *Daily Meeting*, on cadascú dona una breu actualització del seu progrés i, en tot cas, exposa els fets que el bloquegen de poder acabar la seva tasca. Un altre cop, l'*Scrum Master* és l'encarregat de portar la reunió, aconseguir tota aquesta informació, i fer amb ella el més convenient per tal que tothom pugui assolir els seus objectius.

Finalment, després d'un *Sprint* de treball, el més important és comprovar els resultats i veure quines coses s'han fet bé i quines malament. Per això comptem amb les dues últimes reunions que tanquen el *Sprint*: el *Sprint Review* i el *Sprint Retrospective*.

⁶Poker planning: activitat per fer més dinàmiques les estimacions de les tasques. Consisteix en que tots els assistents tenen una petita baralla de cartes amb les diferents estimacions possibles. Aleshores decideixen i mostren tots alhora la seva elecció.

La primera de les dues (*Sprint Review*) serveix, com indica el seu nom, per revisar la feina feta duran l'*Sprint*. Consisteix en una petita *demo*⁷ on cadascú ensenya el resultat de les tasques desenvolupades. Així, el *Product Owner* pot comprovar si compleix els seus requisits, si es podria millorar alguna cosa, o si directament el plantejament principal no és el que realment es volia. Per exemple, si sortissin noves tasques per a fer, es podrien afegir al *Backlog* i, si calgués, incloure-les en la pròxima iteració. Per altra banda, l'*Scrum Master* ha d'analitzar si aquestes tasques realment han tingut la complexitat que havien acordat al principi, si el treballador ha requerit ajudes externes i altres problemes subjacents. Aquesta última feina és important perquè, si es té en compte en futurs *Sprints*, millorarà molt la velocitat⁸ de l'equip.

En canvi, la segona no té per què estar relacionada amb la feina feta. Si n'agafem la traducció literal, "*retrospectiva*" significa "*Que mira endarrere en el temps.*". Amb aquesta reunió el que s'espera és que tots els implicats en aquest *Sprint* analitzin com se'ls ha indicat la feina a fer, els esdeveniments succeïts, les ajudes que han rebut o no, les febleses o punts positius de l'equip, de la planificació, etc. I amb això, donin *feedback* a l'*Scrum Master* per tal de poder millorar alguns aspectes o reforçar-los en funció de les opinions rebudes.

I amb això hauríem conclòs la primera iteració o *Sprint* del nostre projecte. A partir d'aquí tornariem a fer un altre *Sprint Planning* i començar de nou tot el cicle.

Aquest seria el camí proposat pels creadors d'aquest *framework*, però, com hem dit, hi ha equips que prefereixen adaptar-lo a les seves necessitats, ja que a vegades en funció de la mida dels equips, de la complexitat de les tasques a fer o d'altres factors, no és del tot viable seguir al peu de la lletra aquest cicle que hem explicat.

⁷Demo: reunió formada per l'equip de desenvolupament que té com a objectiu mostrar les tasques realitzades durant l'*Sprint*, amb la intenció de rebre'n crítiques sobre aquesta.

⁸Velocitat d'un equip: al final de cada *Sprint*, la suma de les estimacions associades a les històries d'usuari que s'han completat s'anomena velocitat.

4.2 La tècnica Pomodoro - Descobreix què és el temps

Per tal d'explicar aquesta tècnica d'optimització del temps a l'hora de desenvolupar una tasca o un projecte, ens basarem en les paraules del seu propi creador, Francesco Cirillo.

Per posar-nos en context, comencem veient com es va crear aquesta tècnica. A final dels anys 80, Cirillo era un estudiant de sociologia, el qual, com tot altre estudiant, havia de complir amb les seves tasques i exàmens. Preparant-se per un examen, se'n va adonar que li era complicat centrar-se en l'estudi, ja fos per distraccions o perquè li era complicat estar diverses hores seguides desenvolupant la mateixa feina. Aleshores va pensar en repartir-se aquesta feina en diferents espais de temps, deixant entre ells un descans en el qual pogués alliberar un moment la ment i donar-li un descans. Per fer això, va començar a fer aquests temps de focalització durant 2 minuts i descansant-ne un altre. Va veure que aquest mètode no li era efectiu perquè amb només dos minuts no li donava temps a reprendre un altre cop la tasca de l'estudi, així que seguia angoixat i preocupat pel temps que li quedava. Però això el va fer arribar a una conclusió: necessitava aconseguir calma i control sobre la situació.



Figura 4:
Francesco Cirillo

Per aconseguir-ho, es va proposar els següents objectius:

1. Trobar el temps ideal per la durada d'aquests temps de focalització.
2. Esbrinar quantes repeticions en podia fer per tal que li sortís viable sense perdre la concentració.
3. Determinar com fer els descansos entre dos períodes i quina durada haurien de tenir.

També, va veure que això en el que estava pensant li podria ser molt útil pels seus estudis, però que realment es podria aplicar des del dia a dia fins en organitzacions o reunions per tal d'optimitzar el temps tant individual com col·lectiu.

I amb això va començar a fer els primers experiments per assolir els seus objectius. I per això necessitava algun temporitzador, algun element visible que li pogués servir tant per les seves investigacions com per aplicar aquesta nova tècnica. El que li va quedar més a mà va ser un temporitzador de cuina normal i corrent, en forma de tomàquet. I d'aquí va sorgir la tècnica Pomodoro.

Com hem pogut veure, el temps és un factor molt important en aquesta tècnica. Es basa en convertir el temps en el nostre aliat, qui ens ajudarà a complir els nostres objectius i que, en cap cas, ens ha de generar ansietat.

Així que, després de moltes proves, Cirillo va concloure que el millor temps per estar centrat en la mateixa tasca és de 25 minuts, el qual anomenarem a partir d'ara "*pomodoro*". També, com a descans va veure que 5 minuts era suficient per

una persona per tal d'evadir-se i poder tornar a començar el següent període amb més forces que no si l'hagués fet seguit. Però a vegades fer cicles de concentració tan seguits pot acabar per consumir molta de l'energia d'un treballador. Per això, cada quatre pomodoros, aquest quart tindrà un descans de 15 minuts en comptes de 5. Pel que ens quedaria un cicle de $25 + 5$, $25 + 5$, $25 + 5$ i $25 + 15$.

Ara ja només li faltava definir un seguit de regles a seguir per tal d'aprofitar al màxim aquesta tècnica. Entre aquestes, cal tenir en compte les següents consideracions:

- El temporitzador ha de ser visible en tot moment.
- No es pot interrompre un pomodoro.
- Un pomodoro no es pot dividir.
- Si s'interromp un pomodoro passats els 5 primers minuts, cal prendre'l com nul.
- Cal deixar de treballar **sempre** passats els 25 minuts.
- No realitzar tasques mentalment significatives durant els descansos.
- Si la tasca s'acaba abans que el pomodoro, cal seguir amb la mateixa (revisió, repetició, etc.).
- Si la tasca s'acaba en els primers 5 minuts, es pot no incloure aquest pomodoro i començar un altre amb una tasca diferent.
- Una tasca no hauria de durar més de 5 pomodoros.

Aquestes, però, són indicacions donades per Cirillo. Altres estudis i equips han adaptat la tècnica a les seves necessitats, modificant una mica certs aspectes per tal de guanyar rendiment en les seves tasques.

Un cop introduïts en els primers aspectes de la tècnica, podem aprofundir en el cicle de treball d'una persona que la utilitzi. Podem dir, per tant, que es va dividir en 3 etapes:

1. Planificació
2. Seguiment
3. Registre, processament i visualització

Durant la primera, cal que l'individu prengui consciència de la feina que ha de fer, de les tasques que té pendents i del temps que té disponible per realitzar-les.

Prèviament a aquesta planificació diària, és necessari un treball de recollida de tasques per tal de crear un *inventari* de tasques i de la seva prioritat. Aquest inventari definirà el conjunt de tasques a fer per acabar un projecte o treball.

Després d'això, i per començar ara sí, haurem de desenvolupar una planificació de les diverses tasques a fer. Això es pot fer amb qualsevol eina de gestió de tasques, tot i que a vegades un paper i llapis ajuda molt a interioritzar el que volem fer.

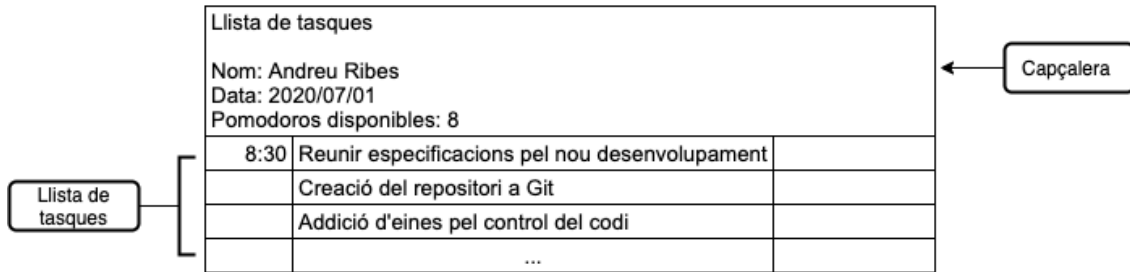


Figura 5: Planificació d'una jornada de Pomodoro

Com podem veure a la *Figura 5*, fer aquesta planificació és ben senzill. Tindrem una petita capçalera amb la data i el nombre de pomodoros disponibles (que pot variar en funció del treballador), i a sota una llista de tasques ordenades per prioritats. Aquestes tasques poden haver estat programades amb antelació, així que si en sabem més informació de quan s'han de dur a terme, podem utilitzar la primera columna de la llista. L'ús de l'última columna vindrà més endavant.

Un cop tenim aquesta planificació, el treballador ja pot començar la seva jornada i el seguiment d'aquesta. Es posarà el temporitzador en marxa i, per tant, comença la primera tasca de la llista.

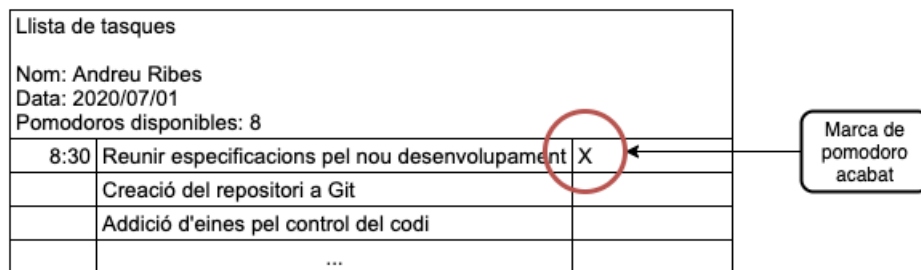


Figura 6: Primer pomodoro acabat

Un cop acabat aquest primer pomodoro, cal que ho marquem amb una creu en l'última columna, tal i com veiem en la *Figura 6*.

Durant el desenvolupament de les tasques, molts esdeveniments poden ocórrer al voltant d'un treballador o, fins i tot, dins del seu cap. Aquests esdeveniments s'anomenen **Interrupcions**.

Evitar aquestes interrupcions és un dels objectius més importants per Cirillo. Molt del temps d'un treballador és perdut per culpa d'aquestes. Per això és important conèixer-les, analitzar-les i, sobretot, minimitzar-les. Podem dividir les interrupcions en:

- Interrupcions internes: pensaments irrelevants per la tasca en curs, com la

necessitat de comprovar el correu electrònic, d'anar a buscar alguna cosa per menjar, etc.

- Interrupcions externes: aquelles que són produïdes per agents externs com, per exemple, un company preguntant, la teva parella fent plans per sortir a sopar, etc.

Quan patim una interrupció, ja sigui interna o externa cal sobretot **protegir el pomodoro**. Amb això ens referim a fer tot el possible per no haver d'anular-lo.

Per tant, primerament cal que fem visible aquesta interrupció. Per fer-ho, afegirem un apòstrof a la columna de pomodoros acabats de la tasca en curs.

Llista de tasques		
Nom: Andreu Ribes		
Data: 2020/07/01		
Pomodoros disponibles: 8		
8:30	Reunir especificacions pel nou desenvolupament	X''
	Creació del repositori a Git	
	Addició d'eines pel control del codi	
	...	

Interrupcions

Figura 7: Visualització de les interrupcions

Tot i això, a vegades aquestes interrupcions són rellevants pel desenvolupament de la nostra tasca o algú altre ens necessita amb urgència per alguna altra. Per això, un cop marcada la interrupció, cal escollir què volem fer amb ella. Les opcions poden ser diverses, però normalment escollirem una de les següents (ordenades per preferència):

1. Ignorar la interrupció: molts cops no és important atendre aquests fets externs i el millor és seguir amb la nostra feina per tal de protegir el pomodoro.
2. Postposar la interrupció: la podem apuntar en una subllista de tasques dins la planificació per tal de tenir-la en compte per més endavant.
3. Anular el pomodoro actual: tot i que hem d'intentar que no passi, alguns cops és necessari anular el pomodoro i canviar de tasca. Això ho marcarem amb un guió en comptes d'una creu.

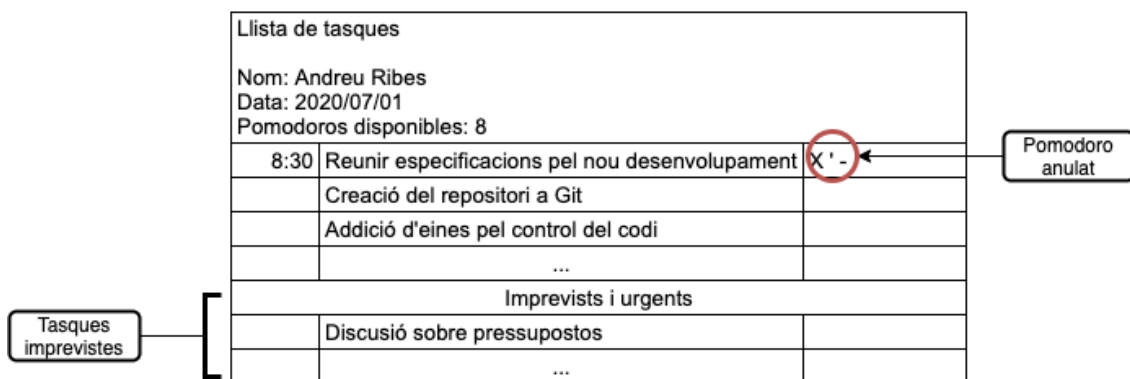


Figura 8: Gestió de les interrupcions

Com podem veure plasmades en la *Figura 8*, aquestes serien les diverses formes de gestionar una interrupció, ja sigui interna o externa. El més important, però, és seguir la política de **protegir el pomodoro**, tal i com ens encomana Cirillo.

Un cop ens habituem a aquesta tècnica i a evitar o controlar les interrupcions, un bon avenç és aconseguir estimar correctament les nostres tasques. Això ens ajudarà molt a programar la jornada i a poder donar aproximacions del que trigarem a fer una feina. Tenint en compte que tenim un nombre màxim de 5 pomodoros per tasca, és important saber agafar una tasca i poder-la partir en cas que es necessiti. Com també cal veure que si una tasca es pot acabar en menys d'un pomodoro, el millor és ajuntar-la amb una altra per tal d'aprofitar aquest pomodoro al màxim.

Per tant, un cop estiguem planificant la llista de tasques a fer, marcarem aquesta estimació amb un seguit de requadres (tants com pomodoros estimem).

El funcionament serà el mateix que abans, però marcarem dins els requadres amb creus aquells pomodoros que acabem. A vegades, però, no encertarem les nostres estimacions i ens podem quedar curts. En aquest cas, podem fer una segona estimació, si ho veiem convenient, mitjançant cercles.

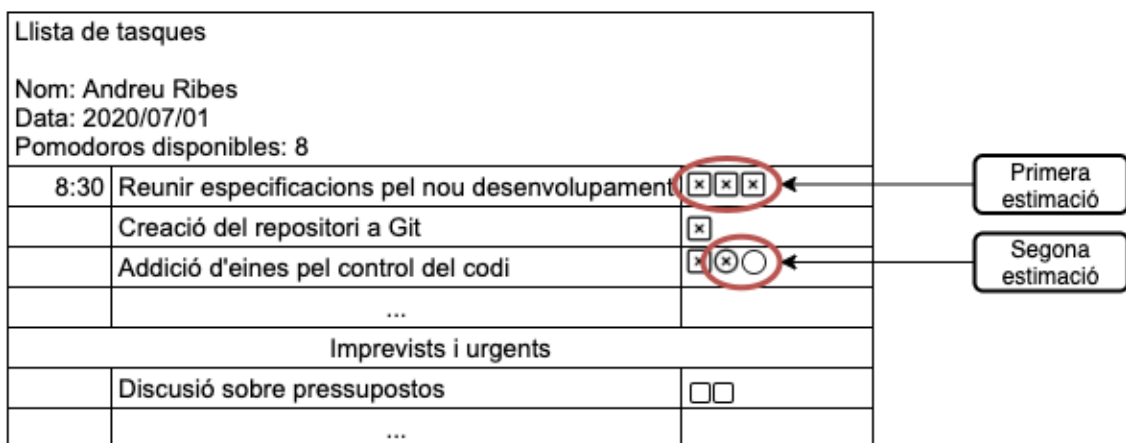


Figura 9: Estimació de pomodoros

Finalment, un cop acabada la jornada, cal que parem un moment i mirem tot el

que hem fet durant el dia. Així podrem analitzar els aspectes bons i dolents, per tal de reforçar-los o corregir-los si s'escau. Aquesta *retrospectiva* també l'hem vist abans quan explicàvem *Scrum*, i per tant podem afirmar que és una peça important de qualsevol metodologia de treball.

En aquest registre inclourem:

- Les tasques a desenvolupar.
- Els pomodoros estimats. En cas de segona estimació ho escriurem com *primera estimació + segona estimació*.
- Els pomodoros reals que hem trigat a acabar la tasca.
- La diferència entre els pomodoros estimats i els reals.

Registre			
Nom: Andreu Ribes			
Data: 2020/07/01			
Tasca	Estimats	Reals	Diferència
Reunir especificacions pel nou desenvolupament	3	3	0
Creació del repositori a Git	1	1	0
Addició d'eines pel control del codi	1+2	2	-1
...	...	...	...

Figura 10: Registre de pomodoros

Hem d'intentar que la diferència s'acosti el màxim a 0 possible. És normal que al principi d'utilitzar la tècnica ens equivoquem amb les nostres estimacions, però justament el registre ens serveix per veure aquests errors i poder ajustar els nostres càlculs en el futur.

Amb això hauríem cobert la funcionalitat principal de la tècnica Pomodoro. Aquesta, però, també es pot estendre a gestió del temps en equips. Però aquesta part i altres de relacionades no són rellevants per a la nostre investigació. Per això, deixem a l'elecció del lector si aprofundir en aquests camps. Un bon començament seria la lectura del llibre de Cirillo, el qual hem utilitzat per a aquesta síntesi de la seva tècnica.

5 Disseny

5.1 La nova metodologia

Just arribem a la part més important d'aquest estudi: la creació d'una nova metodologia híbrida, entre Scrum i Pomodoro, que ens permeti maximitzar el rendiment no només d'un equip sinó també de cadascun dels seus membres.

Per això hem pres aspectes de cadascuna de les parts i les hem variat una mica per tal d'unir-les el millor possible. I per explicar aquest procediment primer explicarem quines parts hem agafat i després com les hem ajuntat per tal de generar aquesta nova hibridació.

En la part d'Scrum hem agafat els següents aspectes:

- Distribució de rols: *Product Owner*, *Scrum Master* i *Scrum Team*.
- Mètode àgil com a base: basarem la nostra metodologia en la iteració mitjançant *Sprints*.
- Les diverses fases del cicle de Scrum: adquisició de requisits, *Sprint Planning*, *Daily Meetings*, *Sprint Review* i *Sprint Retrospective*.
- Estimació de la complexitat de les tasques a fer durant el Sprint.

D'altra banda, de la part de la tècnica Pomodoro ens hem quedat amb:

- Distribució del temps en pomodoros de 25 minuts.
- Les regles bàsiques que cal complir en cada pomodoro (no es poden dividir ni interrompre, deixar de treballar quan sona el temporitzador, fer descansos entre pomodoros, etc.).
- Cicle de la jornada laboral: planificació, seguiment i registre.
- Minimitzar interrupcions (internes i externes) i seguir la política de protegir el pomodoro.
- Estimació en pomodoros de les tasques a complir.

Com podem veure, la unió de ambdues parts no implica cap complicació tècnica. Podríem veure que Scrum seria la capa *externa* de la metodologia, aquella que controla la feina a nivell d'equip i ho fa en intervals definits pels Sprints. En canvi, la tècnica Pomodoro quedaria a un nivell més baix i controlaria el dia a dia dels membres de l'equip, tant treballadors com directius.



Figura 11: Cicle de la metodologia híbrida

Si ens fixem en les dues anteriors llistes, l'únic punt que tindrien en comú és l'estimació de les tasques. Per solucionar això hem decidit que aquesta estimació es faria quan ho estableix Scrum, en el *Sprint Planning*, però es faria amb la mesura de pomodoros, tal i com es fa en la mateixa tècnica. Per tant, durant la reunió, els membres de l'equip decidirien quants pomodoros caldrien per tal d'acabar cada tasca. Es podria utilitzar algun mètode com el *Poker Planning* per tal de realitzar aquestes estimacions. Una situació amb la que ens podríem trobar és que per a un membre una tasca s'acabi en 1 pomodoro, mentre que per un altre la tasca en requereix almenys 3 per finalitzar-se. En aquest cas, és molt més rentable assignar-li la tasca al membre que menys pomodoros necessiti per acabar-la (sempre que sigui possible). Així, sempre i quan les seves estimacions siguin acurades, sabrem que aquesta tasca serà acabada en el menor temps possible.

Ara que ja coneixem per sobre els aspectes que inclourà aquesta nova metodologia, creiem que el millor per conèixer els detalls d'aquesta és explicar un cicle complert del seu funcionament, el qual inclogui el màxim de situacions possibles que ens puguem trobar durant el seu transcurs. En aquesta explicació inclourem nomenclatura ja mencionada en cadascun dels subapartats de l'apartat 4, com també explicacions d'algunes parts, per tal de que el lector no perdi cap detall del que està passant.

5.1.1 Un cicle de la nova metodologia

Començarem suposant que partim de la situació on ja tenim un seguit de requeriments pel nostre projecte o desenvolupament. Per tant, tindrem un *Backlog* amb certes tasques per complir. A partir d'aquí, i com veiem en la *Figura 11*, anem a seguir el cicle típic de l'Scrum.

5.1.1.1 Sprint Planning

Primerament tindríem l'*Sprint Planning*. Aquest, format per l'*Scrum Master* i l'*Scrum Team* decidiran quines tasques entraran en aquest proper cicle. Començaran a agafar-les del *Backlog* en funció de la seva prioritat i, una a una, les aniran estimant. Aquí trobem la primera curiositat de la metodologia. Com hem dit anteriorment, l'equip ha de decidir en consens quants pomodoros requereixen les tasques a finalitzar. Donarem un parell d'exemples per fer més visual les diferents situacions en les que ens podem trobar durant el transcurs d'aquesta activitat:

- Primera tasca: l'equip vota i surten diferents puntuacions, tot i que molt semblants. L'*Scrum Master* decideix que, ja que són semblants, es quedarà amb la més alta per assegurar que aquesta tasca la pugui desenvolupar qualsevol dels que han votat.
- Segona tasca: l'equip torna a donar la seva opinió i es veuen estimacions molt diferent (uns diuen 2 pomodoros i altres en diuen 5). Aquí, al veure una gran diferència, primerament caldria discutir el perquè d'aquestes estimacions. Llavors ens podem trobar en dues situacions:
 1. Al final tots els membres veuen que potser havien estimat malament i s'acaben posant d'acord. Aleshores tornaríem a tenir un exemple com la primera tasca.
 2. Aquesta diferència ve donada pel nivell dels membres. Llavors una opció a tenir en compte seria assignar aquesta tasca a qui la ha estimat amb menys pomodoros, per així optimitzar el temps que aquesta requereix.
- Tercera tasca: després d'una altra votació, veiem que tothom li dona una puntuació de menys d'un pomodoro. Per tant, i intentant seguir amb la filosofia de no dividir els pomodoros, el millor serà veure amb quina altra tasca la podem unir i fer una estimació aleshores de la unió de totes dues.
- Quarta tasca: finalment, com a últim exemple, ens podem trobar en la situació que la mitjana d'estimacions estigui per sobre dels 5 pomodoros. Per tant el que cal fer es analitzar la tasca i subdividir-la en tantes com calgui per tal que cadascuna de les subtasques no superi aquest límit de 5 pomodoros.

Tots els membres de l'equip tenen un nombre de pomodoros que poden realitzar a la setmana. Per exemple, si un membre té disponibles 8 pomodoros cada dia i un *Sprint* té una durada de 10 dies (dues setmanes menys els caps de setmana), aleshores caldrà fer el càlcul i obtenir quants pomodoros té lliures en tot aquest cicle: $8 * 10 = 80$ pomodoros. Per tant, un cop acabades les estimacions, cal que l'*Scrum Master* s'asseguri que tothom té prou tasques per tot aquest temps i que és realista que les puguin finalitzar. Haurem creat, aleshores, un inventari de tasques per a cadascun dels individus. Tot i això, sempre poden sorgir imprevistos durant el transcurs d'un *Sprint*, per això és important aprendre d'aquests i així poder fer millors estimacions pel següent cicle.

5.1.1.2 El dia a dia de l'equip

Seguidament, i després de la planificació prèvia a l'*Sprint*, ens trobem en el seu transcurs. Cada membre de l'equip sap quines tasques ha de fer i quants pomodoros té per fer-les. Suposarem que ens posem en el paper d'un membre de l'equip i anirem pas per pas veient què hauria de fer durant un dels dies de l'*Sprint*. Per no trobar-nos en la situació inicial on encara no s'ha començat cap tasca, també podem suposar que ja ha passat més d'un dia des de l'*Sprint Planning*.

Una de les primeres coses a l'inici de la jornada és la planificació de les tasques que farem durant aquesta. Tal i com explicàvem en l'anàlisi de la tècnica Pomodoro, crearem una llista semblant a la de la *Figura 9*, on tindrem les tasques a fer juntament amb les estimacions que es van fer durant l'*Sprint Planning*. Com tindrem un inventari de tasques on hi haurà totes les nostres tasques a realitzar durant l'*Sprint*, agafarem per ordre de prioritats tantes com puguem per tal d'omplir la nostra quota de pomodoros diaris.

Un cop acabada aquesta planificació, la qual no hauria de durar més d'un pomodoro, serà l'hora de la reunió diària, el *Daily Meeting*, on tot l'equip es reunirà amb l'*Scrum Master* per explicar el progrés en les seves tasques. En un *Daily Meeting* normal d'*Scrum*, cada membre de l'equip explicaria breument el seu progrés i remarcaria si té algun agent bloquejant en les seves tasques. Nosaltres però, hem modificat una mica aquesta reunió per tenir dades més precises sobre cadascun dels integrants de l'equip. No només ens importarà saber si el transcurs de la feina és el correcte o si hi ha algun factor que bloquegi el seu desenvolupament, sinó també si hi ha hagut diferències importants en les estimacions dels pomodoros que hem fet anteriorment. Aquesta informació pot ser clau per l'*Scrum Master*, ja que ens podríem trobar en una de les següents situacions:

- L'estimació no ha estat correcta o no s'han tingut en compte factors que poden influir en la tasca: aquesta situació es dona sovint i és degut a la falta d'experiència a l'hora d'estimar una tasca. Gràcies a aquesta alerta, podem saber que una tasca d'aquest tipus pot portar certs problemes col·laterals, però també podem saber quant temps cal per resoldre'ls, ja que tindrem aquesta diferència de pomodoros. En futures estimacions, seria desitjable que aquest error no tornés a passar i en minimitzéssim aquesta diferència.
- La persona no està rendint com hauria: ja que el control de pomodoros és bastant estricte, una gran diferència pot significar també que el treballador té algun problema a l'hora de desenvolupar les seves tasques. A vegades no és només que una persona sigui vàlida o no per fer la feina, sinó que altres motius externs poden estar afectant en la seva jornada. Un bon *Scrum Master* ha de saber identificar aquestes situacions, avaluar quin pot ser el problema i reportar-lo al personal corresponent perquè se'n faci càrrec.

Amb això, per tant, tindrem molt més control sobre el que cadascun dels individus de l'equip està fent. Tot i que tenir un límit més estricte pot ajudar com a incentiu, cal però donar una sensació de certa llibertat. Qualsevol persona pot

tenir un mal dia i tenir un desequilibri en els seus pomodoros. Per tant, en moments de baix rendiment el més important és aportar motivació i ajuda als que més ho necessitin.

Després d'aquesta reunió, que igualment ha de tenir una curta durada, ens tornem a centrar en un sol treballador. Seguirem amb les nostres tasques seguint els pomodoros establerts i els seus descansos corresponents. No és estrany, tampoc, que algú faci un descans més llarg del que li toca, per necessitat o qualsevol altre motiu. Un punt important aquí és reorganitzar-se la planificació en cas de necessitat i tornar al flux normal de treball el més aviat possible.

Per acabar, després de finalitzar amb totes les tasques que teníem assignades, només ens quedarà omplir el registre amb els pomodoros que hem trigat en fer cadascuna d'elles.

Amb això hauríem conclòs un dia de l'*Sprint*. Com podem veure, és molt senzill acoblar la tècnica Pomodoro amb Scrum, i el parell d'activitats extra que ens requereix no treuen gens de temps comparant amb tot el que guanyem amb l'ús del Pomodoro.

A partir d'aquí hauríem d'anar repetint aquest petit cicle fins que arribem al final de l'*Sprint*.

5.1.1.3 Sprint Review

En aquest moment tots els membres de l'equip haurien de tenir acabades les seves tasques i haurien de ser capaços d'ensenyar la feina feta durant aquest temps. Perquè l'*Sprint Review* no va gaire més enllà d'això.

Com ja vam explicar durant l'anàlisi d'Scrum, es reuniran (com a mínim): el *Product Owner*, l'*Scrum Master* i l'equip. Aleshores veuran si la feina acabada compleix amb els requisits inicials, cosa que ha de comprovar el *Product Owner*, i si es considera que alguna d'aquestes funcionalitats o característiques es pot millorar o polir d'alguna manera. Totes aquestes petites coses es poden anar afegint a la llista de requisits de l'equip, d'on més endavant es tornaran a treure tasques per assignar en futurs *Sprints*.

Tot i que l'explicació d'aquesta part és breu i senzilla, aquesta no és una reunió curta. Normalment aquestes reunions poden portar hores i, a vegades, calen diverses reunions ja que els equips són massa grans.

Però amb aquesta aportació de temps, s'aconsegueixen moltes coses com: saber l'estat del producte final, trobar futurs errors abans que que passin, estalviar temps als treballadors si es veu que una de les parts implementades realment no es vol en el producte, etc.

5.1.1.4 Sprint Retrospective

Finalment ens trobem en l'última part del cicle de la nostra metodologia. Aquest, gens lluny de l'original proposat per Scrum, té la intenció de recaptar totes aquelles

inquietuds o coses ben fetes que tot l'equip ha anat veient durant el transcurs de l'*Sprint*. És important remarcar no només els aspectes negatius sinó també els positius, ja que així sabrem cap a on apuntar a l'hora de planejar estratègies futures d'equip des d'un nivell més organitzatiu.

Per posar un exemple de cada:

- Negatiu: un dels membres de l'equip es queixa que durant unes certes hores on la majoria de gent està treballant, hi ha massa soroll a l'oficina. Això podria constar com una interrupció externa i, per tant, cal trobar-ne una solució per tal que no afecti al desenvolupament de les tasques. Es podrien proposar solucions com, per exemple, habilitar una zona apartada de la de treball on es pogués parlar sense molestar als companys.
- Positiu: l'*Scrum Master* remarca que les estimacions d'aquest cicle han estat molt encertades i gairebé no ens hem desviat, pel que hem complert perfectament amb l'objectiu de l'*Sprint*. Aquest fet motivarà molt a l'equip de treball i entraran molt més motivats al següent.

Un cop acabada la retrospectiva, podríem donat per acabat aquest *Sprint*. Després només cal tornar a començar amb la següent planificació per poder continuar amb aquesta nova metodologia. I així seguiríem, com totes les metodologies àgils, fins a la conclusió del projecte i la seva entrega.

Com hem vist, aquesta unió subtil ens pot aportar molt de control en aspectes que queden fora de l'abast d'un Scrum convencional, així com una gran millora del rendiment dels treballadors i, en conseqüència, una finalització més ràpida del projecte.

Normalment, però, aquestes metodologies no es poden dur a terme fàcilment sense l'ajuda d'una eina externa que ens faciliti una mica el seguiment de totes aquestes dades importants. Per això, i com explicarem a continuació, hem elaborat una aplicació per tal de cobrir aquestes necessitats bàsiques de seguiment, anàlisi i registre.

5.2 L'eina

Avui en dia hi ha un gran nombre d'empreses que desenvolupen aplicacions per facilitar el control del desenvolupament d'un projecte. Algunes, com *Atlassian*⁹, tenen un gran paquet d'aplicacions que s'adapten perfectament a les metodologies més utilitzades com Scrum, Kanban, Waterfall, etc.

Nosaltres, ja que hem elaborat una metodologia que gairebé no havia estat contemplada fins ara, hem pensat que el millor seria construir també una aplicació a conjunt amb aquesta, que ens permeti poder planificar, donar un seguiment i obtenir certs resultats de la feina dels equips que la utilitzin.

En aquest apartat explicarem el procés de creació d'aquesta aplicació, no només a nivell tècnic (models, funcionalitats, etc.) sinó també la part de l'experiència de l'usuari¹⁰ (UX), que creiem que és molt important a l'hora de desenvolupar una aplicació que serà utilitzada per un gran nombre de persones i equips.

5.2.1 Requisits

Igual que a l'inici de qualsevol projecte, el primer que necessitem és saber quin és l'objectiu d'aquesta aplicació i quins requisits funcionals haurà de complir. Sabem que aquesta llista es podria ampliar i podria arribar a ser de grans dimensions, però ens hem centrat en els aspectes essencials de la nostra metodologia i hem intentat cobrir-los el millor possible. Tot i això, sempre quedaran funcionalitats o millores per a un treball futur.

També, per facilitar una mica el desenvolupament i fer més clar el model de dades, hem ajustat l'abast d'aquest per cobrir únicament les necessitats d'una sola empresa, és a dir que no crearem un servei que puguin utilitzar diverses empreses, sinó un *software* que pot utilitzar una empresa de forma interna.

Per tant, i com acabem de dir, anem a veure quins serien els requisits mínims per la nostra aplicació de suport a la metodologia:

- Gestió d'usuaris i rols: possibilitat de crear usuaris amb diferents rols (administrador i usuari) perquè puguin accedir a diferents parts de l'aplicació o tinguin més o menys permisos per dur a terme segons quines accions.
- Creació de tasques: necessitem poder crear tasques per assignar-les als treballadors i guardar-ne informació relacionada:
 1. Títol de la tasca.
 2. Descripció - informació detallada de la tasca.
 3. Estat - una tasca pot estar no començada, en progrés o acabada.

⁹Atlassian: empresa dedicada al desenvolupament de productes per al desenvolupament de programari, la gestió de projectes i de continguts.

¹⁰Experiència de l'usuari (*User Experience*: estil de disseny que es basa en l'experiència de l'usuari a l'hora d'utilitzar l'aplicació final. Per això es centra en les seves necessitats, habilitats i limitacions.

4. Usuari qui té assignada la tasca.
 5. Altre informació relacionada (projecte, classificació, etc.).
- Gestió de tasques: relacionat amb el punt anterior, necessitem una secció on sigui possible editar aquestes tasques o eliminar-les, sempre i quant l'usuari en tingui permís.
 - Visualització ràpida de les tasques assignades per a l'*Sprint*.
 - Seguiment de l'activitat relacionada amb la tècnica Pomodoro:
 1. Poder seleccionar una tasca i començar el pomodoro corresponent.
 2. Visualització del nombre de pomodoros invertits en una tasca.
 3. Vista del temporitzador.
 4. Alerta del temporitzador quan acabi.
 5. Marcador del nombre de pomodoros acabats durant el dia.
 - Analítiques que ens donin informació sobre el rendiment de l'equip:
 1. Gràfica individual de cada usuari per veure la diferència entre pomodoros estimats i pomodoros reals.

Ara que ja tenim aquests requisits, el primer que hem de fer és dissenyar aquesta aplicació seguint les indicacions anteriors. Un dels punts més importants serà la informació que emmagatzemem en l'aplicació, ja que serà aquesta la que ens permeti el fàcil ús de la metodologia i un bon accés als seus resultats. Per això, començarem explicant el model de dades de l'aplicació i després ens endinsarem en el disseny de la vista. A partir d'aquests dos punts, ja podrem seguidament implementar la nostra aplicació.

5.2.2 Model de dades

Per tal de veure en detall aquest model de dades, podem consultar una imatge global de com està format en l'annex corresponent, però en aquest apartat anirem part per part explicant les decisions preses per generar-lo.

Com podem veure en l'annex mencionat fa un moment, el model queda dividit en diverses parts:

- Una primera part dedicada al Scrum: *Tasks, Epics, Projects i Sprints*.
- Una segona que consta la part de la tècnica Pomodoro: *Tasks i Pomodoros*.
- Finalment la part de gestió d'usuaris: *Users*.

És evident que la taula *Tasks* ha estat repetida, però ha sigut intencionadament. Amb això volem reiterar que el punt d'unió entre les dues parts de la metodologia (Scrum i Pomodoro) són les tasques.

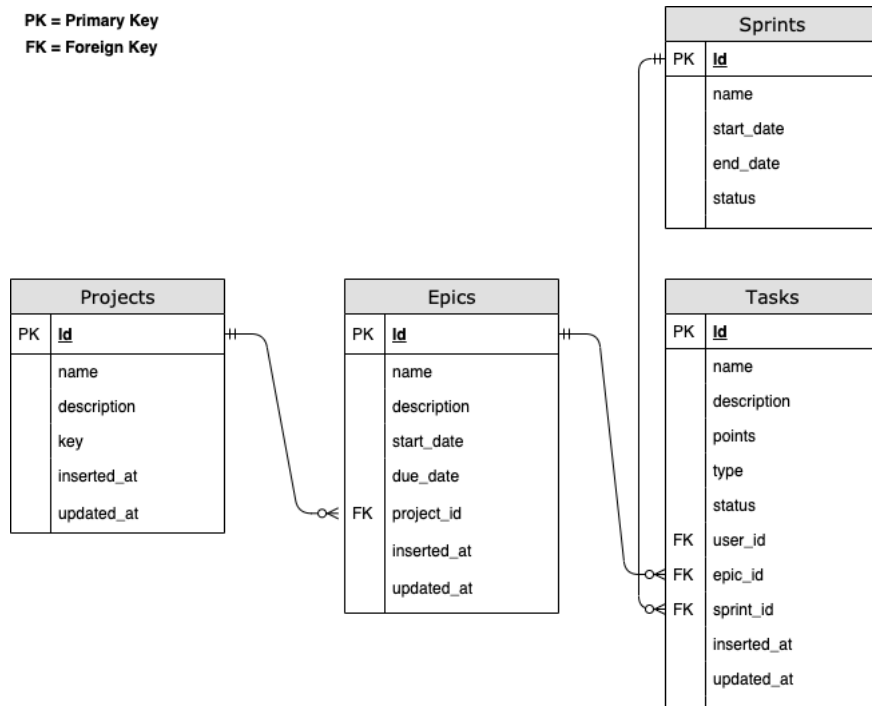


Figura 12: Model de dades - Scrum

Començant per la primera part i, com podem veure a la *Figura 12*, guardarem la informació associada a Scrum de les tasques. Seguint els requisits, dins d'una tasca cal que en guardem la informació bàsica com el nom o la descripció, però també cal saber-ne aspectes essencials per la metodologia com:

- L'*Sprint* en el que s'ha assignat.
- El seu estat: *per començar*, *en progrés* o *acabada*.
- L'estimació de pomodoros associada (entre 1 i 5).
- El tipus de tasca que és: Història d'usuari (*User Story*), Tasca (*Task*) o Error (*Bug*). Se'n podrien definir més, però per simplificar hem optat per aquests 3.

Si ens fixem, aquest model de Tasca té altres models associats mitjançant claus estrangeres o *foreign keys*¹¹.

El primer que trobem és *Sprint*, que ens definirà un cicle mitjançant un nom, les dates de començament i acabament, i un estat que pot ser: *no començat*, *en progrés*

¹¹Clau estrangera (*Foreign Key*): representa l'associació d'un model amb un altre mitjançant la seva clau primària, és a dir el seu identificador en la base de dades.

o acabat. Això ens permetrà situar les tasques en el temps i saber quan s’han fet o, com a mínim, quan es pensava fer-les.

Després, trobem *Epics*. El model de les èpiques¹², utilitzat per altres eines de gestió de tasques, ens permet organitzar les tasques per funcionalitats o seccions. Per exemple, en un projecte que tingui com a objectiu crear una pàgina web de compra-venda d’objectes de segona mà, una èpica podria ser implementar el sistema d’usuaris, i dintre d’aquesta èpica trobaríem tasques com: crear el model de dades adient, crear la vista d’inici de sessió, etc.

Finalment, de la mateixa manera que *Epics*, trobem *Projects*. Els projectes englobaran un conjunt de èpiques i tasques individuals relacionades amb un mateix producte final. Seguint l’exemple anterior, el nostre projecte seria aquesta web de compra-venda d’objectes de segona mà.

En resum, hem creat un model per les tasques que es pugui classificar en dos grups diferents (projecte i èpica) però que alhora es pugui situar en el temps mitjançant Sprints. Tota aquesta informació facilitarà la feina a l’equip en quant a organització i assignació de tasques, però també ens permetrà veure quines parts van més avançades o si s’ha pogut complir tota la feina que s’havia pensat per a un Sprint.

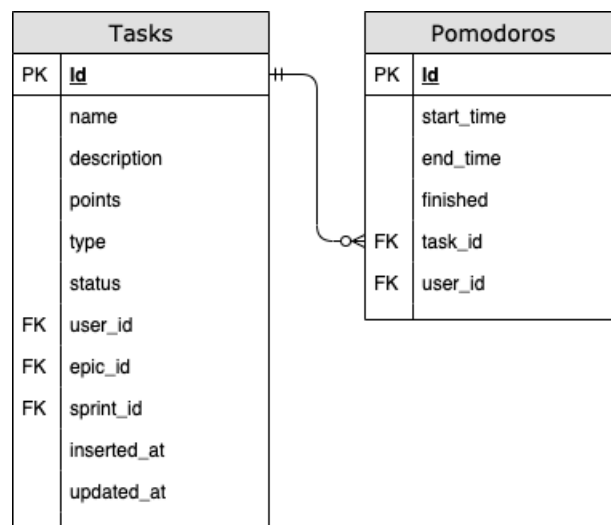


Figura 13: Model de dades - Pomodoro

Un cop explicada la part d’Scrum, passem a l’altra banda de la metodologia. Els models corresponents a la tècnica Pomodoro en són dos: *Tasks*, que l’hem analitzat en l’anterior model, i *Pomodoros*, que ens aportarà tota la informació necessària sobre el rendiment de l’equip i la finalització de les tasques, mentre que obtenim analítiques per poder millorar la nostra estratègia a l’hora d’implementar aquesta metodologia en el nostre equip.

¹²Èpica: gran conjunt de feina que es pot subdividir en diferents tasques. S’anomena d’aquesta manera perquè, en metodologies àgils, aquestes tasques reben el nom de *Històries d’usuari*, i es fa un símil amb la poesia èpica, la qual narra les accions i històries d’un heroi.

Aquest últim model està dissenyat per guardar tres simples coses que també podríem saber si utilitzéssim un temporitzador de cuina:

- En quin moment s’ha començat el compte enrere del temporitzador.
- Quan ha acabat. Pot semblar obvi si establim que un pomodoro dura 25 minuts, però aquesta informació ens serà útil a l’hora de mostrar les analítiques.
- Si el comptador ha acabat o l’hem parat a mitges. Com vam explicar durant l’anàlisi de la tècnica Pomodoro, és importat que el temporitzador sempre acabi, així que aquesta dada ens reportarà si els treballadors tenen massa distraccions i no poden acabar els seus pomodoros.

Tot i això, per acabar d’ajuntar-ho amb la resta del model, ens calen també un parell d’atributs més:

- La tasca associada. Una mateixa tasca pot durar diversos pomodoros, per això ens guardarem un registre per cada pomodoro fet per a aquesta.
- L’usuari que ha fet aquest pomodoro. És important saber qui ha complert o no aquest pomodoro, ja que en podem treure informació com si aquest individu a fet tots els pomodoros que li tocaven durant la jornada o si té alguns problemes que bloquegin la seva feina.

Per tant, ara ja tenim la unió del model entre la part d’Scrum i la de Pomodoro. Amb aquestes taules serem capaços d’obtenir un munt d’informació relacionada amb el projecte i la seva evolució, les seves tasques o l’equip. Aquesta última part, però, necessitarà d’un model propi que veurem a continuació.

Users	
PK	<u>id</u>
	email
	role
	password_hash

Figura 14: Model de dades - Usuaris

Sense treballadors ni un equip, tota aquesta metodologia no tindria gaire sentit. De la mateixa manera, en el nostre model necessitem una taula només per la informació dels usuaris. Aquesta taula, tot i que senzilla, hem vist que es referencia tant en les tasques com en els pomodoros, que tenen un usuari associat.

D’un usuari o treballador en podríem guardar moltíssima informació: nom, cognom, residència, compte bancari, etc. Però per no emmagatzemar informació irrellevant per a aquest projecte, hem decidit en guardar únicament un identificador

(correu electrònic), el rol de l'usuari (administrador o usuari) i una encriptació de la seva clau, amb la qual podrem comprovar més endavant la clau que introdueixin en iniciar sessió, per exemple.

Per tant, i com hem vist, tindrem un model de dades que emmagatzema tota la informació important que es necessita per complir els requeriments bàsics abans esmentats. Encara podríem ampliar o millorar-lo, però és una tasca més adient per a un treball futur.

5.2.3 La vista

Com en la nostra metodologia, Scrum i Pomodoro han de tenir un pes important i hem de fer que el seu ús sigui fàcil i intuïtiu. Però com és gairebé impossible encabir totes les funcionalitats en una mateixa pàgina, hem decidit dividir en diferents seccions cadascuna d'aquestes.

Creiem que una pàgina inacabable tindrà una mala accessibilitat per als nostres usuaris, ja que està demostrat que en la majoria de pàgines on ens podem desplaçar verticalment, la majoria d'aquests no passa de la primera part de dalt de tot. Per això moltes pàgines centren la informació més important en aquesta part. Nosaltres directament no farem una pàgina que tingui moltes parts on ens puguem desplaçar.

Per tant, tindrem un seguit de pàgines el més estàtiques possible, per tal de minimitzar les distraccions dels nostres usuaris, una característica molt apropiada per a la nostra metodologia. Tanmateix, per ajudar en la transició entre pàgines, també comptarem amb un capçal sempre visible.

Anirem una per una per aquestes diferents parts i n'explicarem el disseny i funcionament principal que hem pensat per cadascuna.

5.2.3.1 Capçal de la web

Com acabem de comentar, aquest element, tot i que senzill, donarà mobilitat als usuaris per l'aplicació.



Figura 15: Disseny del capçal de la web

A la *Figura 15* podem veure que aquest element comptarà amb una imatge representativa de l'aplicació, i un conjunt de icones que ens redirigiran a la secció corresponent.

D'esquerra a dreta ens trobem amb els següents accessos:

- Logo de l'aplicació: ens redirigirà a la pàgina principal per no haver d'anar enrere en les altres pàgines.

- **Usuari:** mitjançant aquest, entrarem a la gestió de l'usuari, on podrem realitzar accions relacionades amb el nostre compte de l'aplicació.
- **Analítiques:** ens portarà a la secció d'analítiques, on veurem diferents mètriques dels diversos aspectes controlats per l'aplicació.
- **Desconnexió:** permet a l'usuari actual tancar la seva sessió en la pàgina. Ens portarà directament a la pàgina d'inici de sessió.

Hem decidit posar aquests accessos directes en el capçal ja que creiem que aquestes funcionalitats han de ser molt accessibles per a l'usuari.

5.2.3.2 Inici de sessió i registre

Aquesta pàgina és imprescindible per a l'ús de l'aplicació. Permetrà als usuaris introduir el seu identificador (correu electrònic) i la seva clau. Ambdós paràmetres seran comprovats i, si són correctes, es donarà accés a l'usuari.

Diagrama de disseny de la pàgina d'inici de sessió:

- HEADER**
- Sign in**
- Email**
-
- Password**
-
- SIGN IN**
- [Register](#)

Figura 16: Disseny de la pàgina d'inici de sessió

En la *Figura 16* veiem que tornem a comptar amb el capçal. Aquest, però, només contindrà el logo de la pàgina, ja que, si no s'ha iniciat sessió, l'usuari no tindrà accés a aquestes altres pàgines.

També ens trobem amb una distribució vertical dels elements. Això marca clarament l'ordre en que l'usuari ha d'introduir les seves dades.

Un cop escrites les dades, s'enviaran al servidor mitjançant el botó inferior, on es comprovaran i donaran accés, o no, a la pàgina principal amb la sessió d'usuari corresponent.

Finalment, en el cas d'un nou usuari, tindrem l'opció de registre. Aquesta opció ens redirigirà a una segona pàgina on aquest podrà crear-lo.

HEADER

Register

Email

Password

Password confirmation

REGISTER

Sign in

Figura 17: Disseny de la pàgina de registre

Utilitzant la mateixa distribució que en l'anterior pàgina, en la *Figura 17* veiem que el registre utilitzarà un estil idèntic a la d'inici de sessió. L'única diferència serà l'addició d'un nou camp per confirmar la clau. També, al final de la pàgina, trobarem un enllaç cap a l'anterior pàgina, per tal de donar a l'usuari la possibilitat de tornar en cas d'entrar-hi per equivocació.

Un cop l'usuari es registri correctament, és a dir que les dades introduïdes siguin correctes (correu electrònic i ambdues claus) i que aquest usuari no formi ja part de l'aplicació, la resposta de la pàgina serà la mateixa que en el cas d'iniciar la sessió: ens redirigirà a la pàgina principal.

5.2.3.3 Pàgina principal

Un cop el nostre usuari hagi accedit, ens trobarem amb el nucli de l'aplicació, on la majoria dels usuaris passaran la major part del temps. Hem volgut centrar les funcionalitats més importants en aquesta pàgina per tal que els usuaris no hagin d'anar canviant de pàgina constantment.

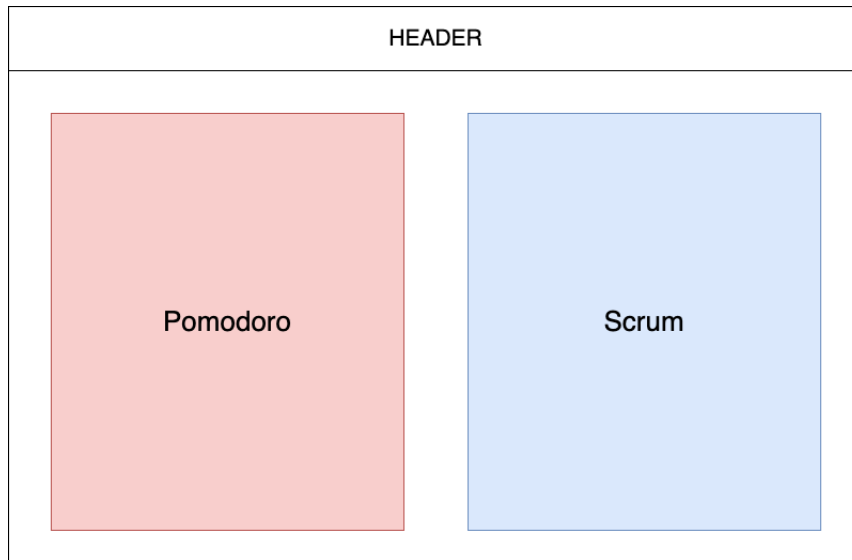


Figura 18: Disseny de la pàgina principal

Aquestes funcionalitats s'ha decidit que es veuran dividides en dues parts, una per les relacionades amb la tècnica Pomodoro, i les altres amb Scrum. En la *Figura 18* veiem clarament aquesta divisió de la pàgina. Tot i això, estaran connectades, de la mateixa manera que en el model de dades, mitjançant les tasques.

Dins de cada una de les parts trobarem objectes amb els quals podrem interactuar per tal d'aplicar la nostra metodologia.

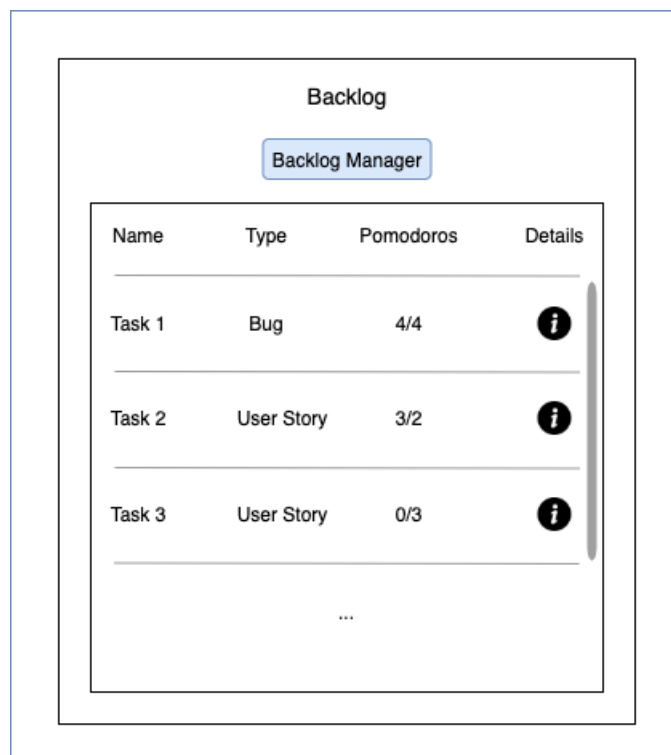


Figura 19: Disseny de la pàgina principal - Scrum (visió com a administrador)

Començant per la secció dedicada a Scrum, trobarem el anteriorment mencionat *Backlog*. Aquest, però, hem intentat que sigui el més útil i pràctic per a tots els diferents rols que tenim. Per això, comptarà amb les següents característiques:

- Mostrarà informació clau de les tasques: nom, tipus, balanç de pomodoros (*fets/estimats*) i un botó que ens permetrà anar al detall d'aquestes.
- Mostrarà només les tasques de l'*Sprint* actual: així no sobrecarregarem aquesta llista amb tasques en les quals el treballador no ha de pensar.
- Mostrarà únicament les tasques per fer o en progrés: la sensació d'anar buidant aquesta llista donarà certa motivació al treballador, alhora que si es veu que aquest fet no passa, pot ser indicador d'algun problema.
- Contindrà un botó per anar al gestor de tasques: aquest serà només visible pels administradors i ens redirigirà a la pàgina corresponent.
- Permetrà desplaçament vertical per navegar entre les diferents tasques.

Aquestes anteriors funcionalitats les podem veure reflectides en la *Figura 19*. Tot i ser una vista relativament senzilla, ens permet accedir ràpidament als detalls de les tasques, obrint-nos la possibilitat de consultar altres atributs com la descripció, la èpica, el projecte, etc.

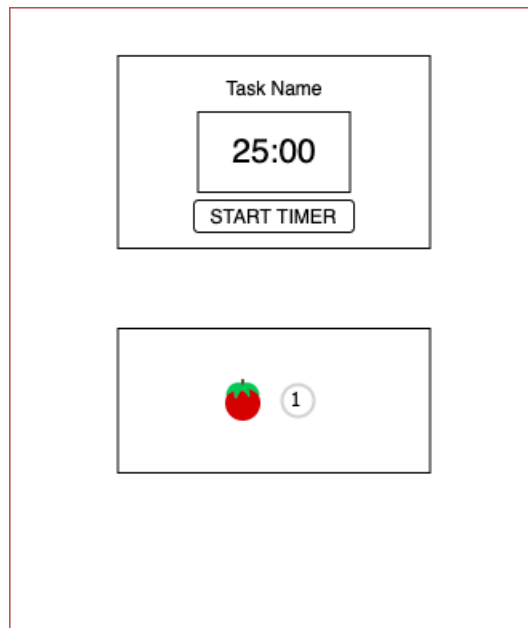


Figura 20: Disseny de la pàgina principal - Pomodoro

Després trobaríem a l'esquerra la part corresponent a la tècnica Pomodoro. Aquesta, igual que la mateixa indica, no ens requereix més que un temporitzador i una llista de tasques que tenim just al costat.

En el temporitzador trobem tres elements:

- El nom de la tasca en curs.
- El temporitzador que anirà disminuint fins a esgotar el temps.
- Un botó que ens permet activar o aturar el temporitzador.

L'ús d'aquest és molt senzill: premem el botó per activar el temporitzador i aquest començarà. En aquest punt, el botó canviarà d'estat i ens permetrà aturar-lo en qualsevol moment abans que s'acabi. Com hem explicat en l'anàlisi de la tècnica Pomodoro, és important no aturar el temporitzador però, en cas que calgués, podríem fer-ho. Això repercutiria en el registre d'aquest pomodoro, que es marcaria com a inacabat. Un cop aquest temporitzador acabi, la pàgina hauria d'emetre una alerta sonora avisant d'aquest fet. Això permet a l'usuari assabentar-se d'aquest esdeveniment per tal de començar el descans.

Finalment, a sota trobem un simple comptador per visualitzar els pomodoros acabats durant el dia. La seva utilitat recau en que el treballador, qui té un nombre de pomodoros disponibles al dia, pugui veure el seu progrés durant el dia. També pot servir com a element motivador el fet de veure que cada cop et queden menys pomodoros per fer.

El més important, però, és la coordinació i comunicació de ambdues parts. Com hem dit durant l'explicació del model de dades, les tasques són el punt mig entre les dues parts. Per això és important fer clar i visible el funcionament conjunt que trobem en la pàgina.

L'únic que haurem de fer per aconseguir això, serà seleccionar una tasca de la llista de Scrum. Aquesta tasca es marcarà com la tasca actual i serà la tasca associada als pomodoros que es comencin des de llavors. En el cas que intentéssim començar un pomodoro sense haver-ho fet, la pàgina ens avisarà que no es pot començar fins que se'n seleccioni una.

Un cop això estigui correcte, el temporitzador començarà el seu compte enrere. En aquest moment està pensat que el treballador comenci o continuï la tasca. És probable que aquest canviï de finestra o realitzi altres activitats fora de l'aplicació. Per tant, és important implementar aquesta funcionalitat de tal forma que l'estat d'aquest temporitzador no es perdi si es surt, per equivocació o no, de la pàgina.

Realment, amb aquesta pàgina un usuari no administrador ja hauria de ser capaç de treballar utilitzant la nostra nova metodologia. Ja que conté tots els elements i funcionalitats bàsiques per fer-ho. Tot i així, abans de poder arribar al punt de realitzar les tasques ens cal introduir-les. Per això ens caldrà un administrador que pugui accedir al gestor de tasques (*Backlog Manager*).

5.2.3.4 Gestor de tasques

Aquesta seria una altra part essencial en la nostra aplicació, ja que ens permetrà gestionar tot allò relacionat amb les tasques. Si seguim el model de dades de la secció anterior, en aquí trobarem les opcions per crear i modificar tasques, sprints, èpiques i projectes.

SPRINTS

EPICS

PROJECTS

←

Filters:

user

▼

epic

▼

Search

NEW TASK

Type	Epic	Name	Status	Points	Sprint	Assignee	Last modification	Details
Bug	User Management	Implement Login	TODO	4	Sprint 1	xrodriguez	2020-07-12	
User Story	Wallet	Implement currency exchange	WIP	3	Sprint 1	csuarez	2020-07-12	
...								

Previous

1

2

3

Next

Figura 21: Disseny del gestor de tasques

Com podem veure a la *Figura 21*, és molt semblant al *Backlog* que trobem a la pàgina principal, però aquest ens mostra molta més informació. A més, no està limitat només a l'usuari actual, sinó que ens permet veure totes les tasques de tots els usuaris.

Podríem dividir la pàgina en dues parts:

- Menú superior: consta de quatre botons de navegació que ens porten als corresponents gestors d'sprints, èpiques i projectes, juntament amb un retorn a la pàgina principal.
- Llista de tasques: una llista amb informació detallada de les tasques, encapçalada per un conjunt de filtres i un botó que ens permetrà crear una nova tasca.

Dins de la part de la llista, hem pensat en afegir la possibilitat de filtrar tasques en funció de l'usuari o la èpica. Un cop escollits els filtres (es poden deixar en buit i no es filtrarà per cap aspecte), podem prémer el boto de cercar (*Search*) i l'aplicació ens retornarà aquesta cerca de tasques filtrades. Un altre característica d'aquesta llista és la paginació. Aquesta ens permetrà retornar resultats més ràpids i lleugers, degut a la baixa càrrega en la resposta.

D'altra banda, el menú de navegació ens dóna la possibilitat de canviar del gestor de tasques al de èpiques, per exemple. Aquestes pàgines són pràcticament iguals

entre si. Els únics canvis que trobarem seran els filtres disponibles i les columnes de la llista, ja que els atributs d'una tasca i els d'un sprint no són els mateixos. Per posar un exemple, podem veure a continuació en la *Figura 22* el gestor de èpiques.

HEADER

←

Filters:

project

▼

Search

NEW EPIC

Name	Project	Start Date	End Date	Details
Wallet	My Wallet App	2020-07-01	2020-11-16	
User Management	My Wallet App	2020-06-16	2020-11-16	
...				

Previous

1

2

3

Next

Figura 22: Disseny del gestor de èpiques

Amb aquestes diferents pàgines, ja serem capaços, des d'un usuari administrador, de poder visualitzar tota la feina feta i per fer de tots els usuaris. Ara només ens quedaria el fet de crear tota aquesta infraestructura de projectes, èpiques, sprints i tasques. Per això comptarem amb els botons que hem mencionat anteriorment i que ens podem trobar a la cantonada superior dreta de les llistes de cada gestor.

5.2.3.5 Creador i editor de tasques

Durant el cicle de la nostra nova metodologia, hi haurà molts moments on haurem d'afegir o editar tasques, projectes, èpiques i sprints. No només durant l'*Sprint Planning*, sinó també en el dia a dia d'aquest sprint, ja que els treballadors que no es dediquin a desenvolupar les tasques ja acordades, sempre tindran més feina aconseguint nous requeriments i establint com aquests s'haurien d'implementar.

Per això, és important comptar amb una pàgina on puguem dur a terme aquesta acció. Evidentment estem parlant d'un o més formularis on introduïrem els detalls d'aquests nous models a guardar en la nostra base de dades.

HEADER	
Epic	None
Name	
Description	
Status	TODO
Points	1
Type	Task
User	None
Sprint	None
SUBMIT	

Figura 23: Disseny del formulari per crear tasques

Aquest senzill formulari el podem veure en la *Figura 23*. Consta dels principals camps descrits en el model de dades per les tasques, i ens permet escollir en diverses llistes desplegable aquelles referències a altres models, com per exemple en el primer camp *Epics*, que si l'obrim veurem una llista d'èpiques per seleccionar. Això enllaçarà, per tant, aquesta nova tasca amb aquesta èpica, i podrem filtrar aleshores aquesta tasca en el gestor de tasques mitjançant aquest enllaç. El mateix cas el trobaríem quan seleccionem l'usuari a qui se li assignarà la tasca, cosa que farem en el camp *User*.

No és obligatori omplir tots els camps, ja que molts es poden deixar en blanc (*NULL*) en la base de dades. Aquest fet s'estén a la resta de models, i els detalls dels tipus dels camps i si es poden deixar en blanc o no, ho podem trobar en l'annex corresponent (*A.1*).

Un cop haguem omplert la informació necessària, només caldrà prémer el botó *Submit* per guardar la tasca. En el cas que ens haguem deixat algun paràmetre obligatori, quan enviem la informació l'aplicació ens retornarà a la mateixa pàgina amb un missatge d'advertència en el camp corresponent.

Aleshores, si aconseguim guardar aquesta nova tasca, la podrem visualitzar en el gestor de tasques que hem vist en la *Figura 21*. Si veiem aquesta imatge, al final del detall de la tasca trobem un botó en la columna *Details*, que ens portarà al detall de la tasca on podrem també editar aquesta informació.

El formulari d'edició serà el mateix que el de creació, només que demanarem primer la informació a l'aplicació per tal que ens ompli els camps amb antelació.

HEADER

Name

Description

Key

Related Epics

Name	Project	Start Date	End Date	
Wallet	My Wallet App	2020-07-01	2020-11-16	i
User Management	My Wallet App	2020-06-16	2020-11-16	i
...				

Figura 24: Disseny del formulari d'edició d'un projecte

També, però, tindrem formularis per crear o editar sprints, èpiques i projectes. L'estructura d'aquests serà molt semblant a la del de les tasques, però amb la diferència que inclourà una llista a la part inferior on podrem consultar els models associats. Per exemple, si estem en el detall d'un projecte, veurem les èpiques relacionades en aquesta llista, tal i com observem en la *Figura 24*.

Amb aquesta pàgina i les anteriors esmentades fins ara, ja podríem implementar gairebé del tot la nostra metodologia híbrida. Podem crear usuaris i entrar a l'aplicació, crear tasques i totes les estructures relacionades, generar nous sprints i associar-ne les tasques que hagin de ser realitzades en ells, etc. Ara, però, l'única peça que ens faltaria és veure com aquesta feina evoluciona i si ho fa a la velocitat esperada o no. Per això comptarem amb l'apartat d'anàlitzes.

5.2.3.6 Analítiques

Des de qualsevol punt de l'aplicació, com hem dit a l'inici de l'anàlisi dels elements de l'aplicació, podrem visualitzar el capçal d'accesos directes d'aquesta. Tornant a la *Figura 15*, del grup d'icones premerem el corresponent a les analítiques.

HEADER

Pomodoro Balance

Filters:

Sprint

▼

User

▼

Search

User	Start Date	End Date	Pomodoro Balance
xrm@mybusiness.com	2020-07-20	2020-07-31	70/62

Figura 25: Disseny de la pàgina d'anàlitzes - Balanç de Pomodoro (visió com a administrador)

Un cop redirigits, ens trobarem en la secció d'anàlitzes de l'aplicació. Com hem dit, aquesta ens reportarà dades rellevants sobre els treballadors i la seva feina.

En la *Figura 25* podem observar una estructura semblant a la del gestor de tasques: uns botons de navegació, un conjunt de filtres, i un resultat a sota en forma de taula. Això ho hem decidit així ja que el fet d'utilitzar pàgines amb estructures semblants facilita molt l'aprenentatge de l'usuari a l'hora d'utilitzar la pàgina.

Per tant, de la mateixa manera que en el gestor de tasques, aquí podem filtrar la informació mostrada en funció de certs aspectes, en aquest cas l'sprint i l'usuari. Aquest últim filtre (*User*), però, serà només accessible per a administradors, ja que creiem que no és necessari que tots els usuaris puguin veure la informació de la resta.

En aquesta primera versió, comptarem amb la següent analítica:

- **Balanç de Pomodoro:** ens indicarà el nombre de pomodoros reals respecte els estimats totals per a una jornada del treballador. Això ens dirà si ha calgut més temps o menys en les seves tasques o, en cas que el nombre de pomodoros sigui baix, que han tingut un gran nombre d'interrupcions. Com vam explicar en l'anàlisi de la tècnica Pomodoro, cal que aquest balanç s'acosti el màxim possible a 0. Si l'encarregat de veure aquests informes (en el nostre cas l'*Scrum Master*) veu un gran desnivell en aquesta gràfica, una intervenció amb aquest treballador podria solucionar el problema.

Podríem elaborar moltes més analítiques i mètriques per a les nostres dades, però creiem que aquesta seria la principal i més necessària a l'hora d'implementar la metodologia. Una altra bona incorporació, per exemple, seria el ben conegut *Burndown Chart*¹³.

¹³Burndown Chart: gràfic consistent en la quantitat de feina restant en funció del temps que ha passat des de l'inici del projecte o cicle.

5.2.3.7 Gestió d'usuari

Finalment, però no menys important, necessitarem un petit espai on l'usuari pugui visualitzar i gestionar les seves dades, afegir-ne, o, en cas que sigui un administrador, canviar els rols dels usuaris.

Per això tindrem l'espai de gestió d'usuaris, al qual hi podem accedir també des de la capçalera de l'aplicació.

HEADER	
User Info	
Identifier: xrm@mybusiness.com	
User Roles:	
Identifier	Role
xrm@mybusiness.com	Admin ↓
csuarez@mybusiness.com	User ↓
...	
Previous 1 2 3 Next	

Figura 26: Disseny de la pàgina de gestió d'usuari (visió com a administrador)

Aquesta pàgina també tindrà diferents aspectes en funció del rol que interpreti l'usuari actual, però en la *Figura 26* podem veure que el seu ús és ben senzill.

La primera part, formada per la informació de l'usuari actual, serà visible per a tots els tipus de rols. En canvi, com comentàvem abans, la llista dels usuaris i l'actualització dels seus rols serà només visible per als administradors.

Si un usuari és administrador i vol canviar el rol d'un altre, des del gestor només ha de seleccionar el rol desitjat en el desplegable i aplicar-lo amb el botó que té a la dreta del tot de l'usuari que està intentant canviar.

6 Implementació

Ara que ja coneixem el disseny i funcionament de l'aplicació, només ens cal implementar-la i provar-la per comprovar si els objectius de la metodologia es compleixen tal i com esperem.

6.1 Entorn de desenvolupament

Per implementar aquesta aplicació, hem decidit utilitzar un llenguatge diferent als normalment utilitzats (Java, Python, etc.). Elixir¹⁴ ha sigut la nostra elecció en aquest cas, i no només per la seva rapidesa a l'hora de gestionar processos, sinó també per la facilitat que ens dóna per crear-ne d'altres per als nostres propòsits. Juntament amb aquest llenguatge utilitzarem el seu *framework* més utilitzat: Phoenix Framework¹⁵. Aquest, que segueix l'arquitectura Model-Vista-Controlador¹⁶, ens permet desenvolupar tant el *Frontend*¹⁷ com el *Backend*¹⁸ de l'aplicació d'una manera relativament senzilla i molt eficient.

Tot i que Elixir ens aporta un gran nombre de característiques realment interessants, deixarem l'opció al lector d'investigar sobre el tema si és que no el coneix ja actualment.

A part del llenguatge de programació que utilitzarem, també necessitarem un lloc on guardar tota la informació de l'aplicació. Per a això hem escollit com a base de dades MySQL, ja que és molt ben coneguda i senzilla d'utilitzar. Per a una primera prova no ens caldran grans optimitzacions ni un gran abast de connexions, així que treballarem únicament amb una instància d'aquesta base de dades.

Aquesta aplicació estarà desenvolupada en un sistema local, pel que comptarem amb un entorn utilitzable en aquest mitjançant ¹⁹. D'altra banda, també estarà preparada per pujar-se a un servidor (en el nostre cas Google Cloud²⁰), amb la finalitat que diferents usuaris puguin utilitzar la mateixa instància de l'aplicació.

¹⁴Elixir: llenguatge de programació funcional i concurrent que s'executa a la màquina virtual d'Erlang.

¹⁵Phoenix Framework: conegut com el *framework* més utilitzat d'Elixir, és utilitzat per elaborar aplicacions web interactives ràpidament.

¹⁶Arquitectura Model-Vista-Controlador: patró de disseny utilitzat per al desenvolupament d'interfícies d'usuari.

¹⁷Frontend: part de l'aplicació visible per l'usuari final.

¹⁸Backend: part de l'aplicació encarregada dels processos interns i gestions amb la base de dades.

¹⁹Docker: programari que automatitza el desplegament d'aplicacions en contenidors de programari.

²⁰Google Cloud: servei de Google que permet muntar infraestructures de programari en servidors dedicats.

6.2 L'aplicació

La nostra aplicació, penjada al GitHub²¹, conté documentació detallada de les seves funcions i models. En aquest apartat no volem esplaïar-nos explicant de nou el funcionament de l'aplicació, cosa que ja hem fet en l'anterior secció del disseny d'aquesta. Per això comentarem els trets més interessants i rellevants que ens ha calgut implementar per tal d'arribar als nostres objectius.

6.2.1 El temporitzador, nucli del Pomodoro

Aquesta seria la part més important de tota l'aplicació, sense la qual no podríem aplicar la metodologia. No entrarem a nivells massa tècnics, però explicarem per sobre com utilitzarem Elixir i els seus processos per tal de crear un temporitzador únic per cada usuari i que no depengui de la vista de la pàgina per funcionar, sinó que el seu estat estarà guardat en el servidor.

Una de les primeres coses que ens vam plantejar a l'hora de crear l'aplicació va ser com desenvolupar aquest temporitzador i que, passés el que passés (excepte una caiguda del servidor), l'usuari pogués tornar a veure'l sense haver de tornar-lo a començar. Ens referim a errors humans com tancar la pàgina, o el fet de poder navegar per altres seccions d'aquesta sense patir perquè el temporitzador falli.

Elixir és un llenguatge que motiva molt a utilitzar els seus processos. Aquests, que poden funcionar de forma concurrent²², ens van inspirar la idea de crear un procés per cada usuari amb l'ajuda d'aquesta característica. Això ho faríem mitjançant un mòdul anomenat *GenServer*. Aquest, com comenta la seva documentació, ens servirà per "mantenir estats, executar codi asíncronament, entre altres coses". Aquí és on vam veure que ens seria realment útil.

Un temporitzador no és més que un compte enrere que disminueix cada 1 segon. Per tant, podem construir-lo justament amb el que acabem d'esmentar:

- Un estat que contingui:
 1. El temps restant.
 2. La tasca en curs.
 3. L'usuari que realitza la tasca.
 4. El temps en que ha començat aquest temporitzador.
- Un conjunt de codi que, seguint les regles del temporitzador establertes per la nostra metodologia, acabi guardant el registre d'aquest pomodoro en la base de dades, tot i que l'usuari decideixi aturar-lo abans de que acabi.

²¹GitHub: servei de control de versions que ens ofereix la possibilitat d'emmagatzemar codi en un servidor extern.

²²Concurrència: fet que dos o més processos o algorismes s'executin alhora sense que el resultat d'un afecti a l'altre.

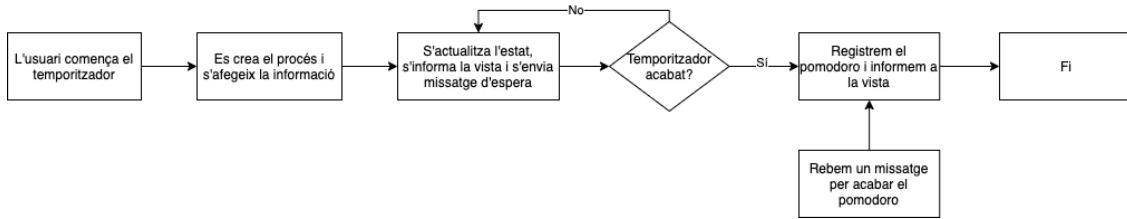


Figura 27: Flux de funcionament del temporitzador

Així que aquesta última part és bàsicament la lògica següent, que també trobem a la *Figura 27*:

- Quan arriba una senyal per part de l'usuari per començar un temporitzador, crea un nou procés que contingui la informació que aquest ens envia (tasca i usuari).
- Quan aquest procés es crei, afegeix-lo al registre de temporitzadors²³ i envia-li un missatge asíncron al cap d'un segon perquè s'actualitzi.
- Quan rebim un missatge d'actualització, resta un segon al temps restant de l'estat, informa a la vista d'aquest i torna a enviar el mateix missatge.
- Si el temps restant arriba a ser 0, aleshores afegeix aquest pomodoro a la base de dades i avisa a la vista d'aquest esdeveniment.
- Si, en canvi, el temps és major a 0 però rebem una cancel·lació del pomodoro, interromp aquest temporitzador i afegeix aquest pomodoro com a *no acabat* a la base de dades.

Com podem comprovar, en aquest procés hem d'avisar a la vista contínuament per tal que actualitzi el temps en pantalla. Per tal de fer això, el que farem és obrir una connexió *WebSocket* amb el client (la pàgina que visualitza l'usuari) i li enviarem un petit missatge perquè actualitzi la informació. Phoenix ens facilita molt la feina aquí amb l'ús d'un altre dels seus mòduls: *Channel*. Cada cop que un usuari entra a la pàgina, es crea directament aquesta connexió amb el servidor. Això ens permet activar accions d'una forma ràpida, com per exemple el fet de començar o aturar el temporitzador, o la rebuda d'aquests missatges amb el temps restant.

Finalment, d'aquest temporitzador podem dir que, tot i que està dissenyat a prova d'errors de l'usuari, no ho està pels errors del servidor. Si per un cas un dels processos que interpreta un temporitzador d'un usuari deixa de funcionar, perdriem l'estat d'aquest i per tant el temps restant i la informació relacionada. Això ens passa en aquesta primera versió, però com a treball futur deixem l'oportunitat de blindar aquest temporitzador amb una estratègia molt simple: cada cop que hi hagi

²³Registre de temporitzadors: el mòdul *Registry* d'Elixir ens permet crear una taula en memòria que emmagatzemarà l'identificador dels temporitzadors per tal d'accedir a aquests processos en un altre moment.

un canvi d'estat, guardar-lo en una taula d'accés molt ràpid (recomanem ETS²⁴ ja que treballem amb Elixir). Això ens permetria recuperar l'estat en cas de la caiguda d'aquest procés. Una solució més forta seria el fet de guardar aquest estat no només en una taula en memòria, sinó també a disc, per tal de recuperar-lo en cas que tota l'aplicació o el sistema deixés de funcionar, ja que en aquest cas perdríem tot el que teníem guardat en memòria.

6.2.2 Models i usuaris

En aquí no hi ha gaire secret. Per elaborar els models ens hem basat en el disseny de la base de dades que en comentat en la secció **5.2.2**. Un altre cop Elixir i la seva comunitat ens faciliten la feina mitjançant una dependència²⁵ que s'utilitza la gran majoria de les vegades per a muntar els esquemes de les bases de dades i les connexions amb aquestes: *Ecto*.

Donat que Elixir és un llenguatge funcional, no construirem mai *objectes*, sinó *estructures*. Aquestes es poden definir i comprovar fàcilment gràcies a *Ecto* i un parell de submòduls d'aquest anomenats *Schema* i *Changeset*, respectivament. En el codi font trobarem tots aquests models que demanàvem en els requeriments implementats mitjançant aquestes eines.

Per als usuaris, com a part del model necessitem certa implementació per tal de poder registrar i donar accés als usuaris, hem utilitzat una dependència que ens dóna gran part d'aquesta feina feta: *Pow*[13]. Alhora, ens deixa ampliar els seus models i funcionalitats si s'escau.

6.2.3 Vista

Finalment, ara que tenim implementades les funcionalitats i són accessibles els models de la base de dades, només ens falta crear les vistes seguint les dissenys que hem elaborat anteriorment.

Per a implementar això també utilitzarem Phoenix. Aquest, mitjançant el seu mòdul *HTML* ens dóna un conjunt d'eines per fer-ho. Per exemple, podem utilitzar funcions internes a aquest per a elaborar camps de formularis i, fins i tot, els botons que ens serviran per a enviar la informació al servidor. D'altra banda, podem implementar funcions que s'executin en la pàgina sense necessitat de fitxers Javascript. Això ho hem utilitzat sobretot a l'hora de demanar informació de la base de dades com poden ser les llistes de tasques, èpiques, etc.

²⁴Erlang Term Storage: sistema d'emmagatzematge en memòria que permet guardar i accedir a informació de forma molt ràpida, gràcies a que està continguda dins de la mateixa màquina virtual de Erlang.

²⁵Dependència: llibreria o conjunt de codi independent a una aplicació que pot ser utilitzat en aquesta sense haver de tornar a implementar les seves funcionalitats.

7 Experiment

7.1 Plantejament

El següent experiment planteja un escenari de comparació entre dues situacions: l'ús de la metodologia híbrida entre Scrum i Pomodoro, o l'ús d'una metodologia Scrum convencional.

La hipòtesi a comprovar és:

Si un equip implementa aquesta nova metodologia com a eina de treball, el seu rendiment incrementarà i, per tant, es trigarà menys temps a obtenir els mateixos objectius sense l'ús d'aquesta.

La durada d'aquest experiment es data en 4 setmanes entre el dia 13 de juliol i el 7 d'agost de 2020. Durant aquest temps es realitzaran 4 sprints de treball (1 per setmana) en els quals s'ha d'implementar aquesta nova metodologia i realitzar totes les activitats relacionades (*Sprint Planning*, *Daily Meetings*, *Sprint Review* i *Sprint Retrospective*). A banda, recollirem informació diària sobre les tasques que cada membre de l'equip realitza i els pomodoros que inverteixen en aquestes.

Degut a la privacitat del projecte en el que aquest equip està treballant, no podrem donar informació exacte de les tasques ni de la feina realitzada. Així que haurem de donar noms de tasques, èpiques i projectes diferents a les reals.

7.2 Persones

Per a aquesta prova tindrem un equip de 4 persones que actualment treballen en un projecte per a una empresa real:

- 1 *Scrum Master* que també exercirà de desenvolupador.
- 3 membres de l'equip que exclusivament treballaran en desenvolupament.

El paper de *Product Owner* que està inclòs en la metodologia no ens caldrà en aquest cas, ja que ja comptarem prèviament amb els requeriments de les tasques.

A aquestes persones, se'ls ha demanat el consentiment informat que podem trobar en l'annex corresponent, amb la finalitat de poder tractar les seves dades i resultats de l'experiment, sempre fent anònimes les seves identitats.

Totes es situen dins d'una mateixa franja d'edat entre els 23 i els 30 anys, pel que els considerem població jove. Tanmateix, les seves habilitats professionals i experiència estan en el mateix nivell aproximadament, pel que no hauríem de trobar enormes diferències de rendiment entre els uns i els altres, a excepció de que tinguin diferents velocitats d'adaptació a la metodologia. També destacar que tots ells treballen 8 hores al dia de dilluns a divendres, i comptarem amb la seva disponibilitat completa per a aquest període de temps.

Finalment cal esmentar que aquest grup realment és un subgrup d'un equip més gran. Com no tot l'equip general estarà utilitzant la metodologia, pot ser que veiem afectats els resultats per culpa d'interrupcions externes.

7.3 Entorn i eines

Degut a l'incident del COVID-19, l'entorn on l'equip desenvoluparà les seves tasques és molt diferent al que normalment trobaríem, és a dir el típic entorn d'oficina. Tindrem que cadascun d'ells treballarà des del seu domicili, on preveiem que les interrupcions no siguin les mateixes. Estimem que les externes disminuiran però les internes augmentaran, ja que el fet de no estar en una oficina i tenir sensació de "no control", a vegades pot provocar aquestes distraccions.

Per a realitzar aquest experiment, comptarem amb una instància de l'aplicació desplegada a Google Cloud, a la qual podran accedir tots els membres de l'equip des dels seus dispositius a través del navegador web.

De la mateixa manera, la base de dades també serà única i estarà també en el servei de Google. Això ens permetrà reunir totes les dades en un sol lloc per a un millor anàlisi posterior.

En quant al temps disponible de cada membre de l'equip, hem calculat que les seves 8 hores de feina es converteixen en 16 pomodoros. Tot i això, hem començat amb una disponibilitat real de 14 per tal de poder compensar inconvenients que es trobin durant les seves jornades. Això no significa, però, que aquests deixin de treballar passats aquests 14 pomodoros. En cas que es compleixin tots i els sobri temps, hi haurà la possibilitat de realitzar-ne més, possibilitant així l'avenç de feina.

7.4 Entrenament

Es realitzarà un entrenament previ per a tots els membres de l'equip, ja que no en són coneixedors d'aquesta metodologia. Tot i això, el projecte en el que es troben ja implementa la metodologia Scrum i tots han sentit a parlar de la tècnica Pomodoro, encara que no en coneixen el funcionament ni la han utilitzat. Per això esperem que l'aprenentatge sigui ràpid i no suposi cap entrebanc.

En aquest entrenament es farà una molt breu simulació d'un cicle, on explicarem els objectius de cadascuna de les activitats involucrades i quina informació han d'aportar per tal que els resultats siguin el millor possibles.

A partir d'aquí, tots els membres de l'experiment haurien de ser capaços de desenvolupar tota la feina de manera autònoma. Igualment se'ls ha deixat obert un canal de xat per a consultes i possibles millores de l'aplicació.

7.5 Primera iteració

Degut a l'anterior utilització d'Scrum en l'equip, aquest ja comptava amb els requeriments i les tasques a desenvolupar per a aquesta setmana. Tot i això, se'n va

realitzar un primer *Sprint Planning* per tal d'estimar-les i començar el cicle tal i com estipula la metodologia. Aquesta reunió va durar aproximadament una hora (relativament curta) i es van fer estimacions bastant altes, amb una mitjana de 3-4 pomodoros per tasca. Aquestes estimacions no van portar gaires discrepàncies entre els diferents membres i se'ls van assignar en funció del projecte en el que cadascú es trobava.

Després va començar la part de desenvolupament, en la qual es van dur a terme 4 *Daily Meetings*. En aquests vèiem certa confiança en els membres de l'equip, les primeres tasques, com ja esperàvem, difícilment seguien sobre els pomodoros estimats, tot i que tothom complia amb els temporitzadors evitant les interrupcions externes. Cap a finals d'aquest primer cicle entraven certes preguntes com: “com evitar les interrupcions” o “què fer en cas d'interrupció urgent”. Després de solucionar-les, però, l'equip tenia una millor idea de com conviure amb la metodologia.

Un cop acabades l'etapa de desenvolupament, vam arribar a les dues reunions finals. La *Sprint Review* ja estava implementada en l'equip, així que tots van ser capaços d'ensenyar la seva feina sense cap mena de problema. En canvi, l'*Sprint Retrospective* la vam fer a part dedicant-li una estona a la metodologia. En aquesta última, vam rebre opinions diverses sobre la metodologia i l'aplicació. Un dels comentaris més destacables va ser “com subdividir tasques que a vegades no són divisibles per tal que càpiguen dins de 5 pomodoros”.

Per tant podem dir que el primer cicle de l'experiment va ser tot un èxit ja que en aquest l'objectiu era bàsicament que l'equip conegués més en profunditat els detalls de la metodologia i el seu funcionament. D'altra banda, vam observar algunes mancances ja esperades com estimacions errònies, interrupcions inevitables”, etc.

7.6 Segona iteració

De la mateixa manera que en la primera, aquesta iteració va ser encapçalada per la planificació. En aquesta cadascú va aportar les tasques amb més prioritat dels seus respectius projectes. Després es va procedir a l'estimació d'aquestes i, en aquest cas, sí que vam observar diferències entre les estimacions dels diferents membres de l'equip. Després d'una curta discussió, en la majoria de casos es va optar per les estimacions més elevades. Creiem que això va ser conseqüència dels desequilibris en les estimacions del primer cicle, que no van ser del tot acurades.

Durant el transcurs d'aquest segon sprint, ens vam trobar amb certs problemes amb interrupcions externes que ja vèiem venir degut al treball remot. Es van donar uns quants casos d'acumulació d'aquestes interrupcions externes que acabaven requerint la participació del treballador. Aquest fet desquadrava la planificació de pomodoros d'aquest i, per tant, augmentava la sensació d'ofegament que acabava en un menor rendiment. També es va preguntar si “es podien canviar les estimacions de les tasques”. La resposta va ser evidentment negativa, ja que això malmetria els resultats de l'experiment i aniria en contra dels principis de la metodologia.

Aquests últims problemes van ser reportats al final del cicle en la retrospectiva. A part d'aquests, l'equip començava a veure més clares les avantatges de l'ús combinat

de ambdues pràctiques, Scrum i Pomodoro. Creien que l'addició de Pomodoro estava aportant un bon increment de la concentració, tot i que encara havien d'acabar d'aprendre a com gestionar aquestes interrupcions abans esmentades.

7.7 Tercera iteració

En aquesta penúltima, l'equip ja tenia interioritzada la metodologia i els temps de les reunions. Durant el transcurs d'aquestes, es notava molta més fluïdesa passant de la estimació d'una tasca a la següent, i això va venir acompanyat d'una millor subdivisió de les tasques i, per tant, una disminució de les estimacions d'aquestes que va voltar la mitjana de 3 punts.

Aquest últim fet va aportar un millor desenvolupament durant aquest cicle. Els balanços eren bastant més correctes respecte el primer cicle però el problema de les interrupcions externes per part de treballadors fora de l'experiment seguia en peu. Per això vam pensar que adaptariem mesures al respecte en el pròxim cicle.

Durant la majoria de reunions diàries no hi va haver gaires problemes que bloquejessin les tasques dels desenvolupadors. Els únics problemes que es van trobar eren dependències entre tasques que vam poder solucionar mitjançant el canvi de prioritats d'aquestes.

Finalment, un altre cop va sortir a la retrospectiva el problema relacionat amb les interrupcions externes. Tot i això, la majoria de desenvolupadors va adaptar una posició conservadora envers la metodologia i deixaven aquestes tasques secundàries en una cua de feina a fer més endavant. Així vam poder comprovar que seguien el principi de *protegir el pomodoro*, com també vam poder veure que la tendència a estimar de forma elevada encara seguia en alguns casos, ja que algunes de les tasques s'acabaven abans d'aquest temps estimat.

7.8 Quarta iteració

Per últim, vam arribar a aquesta darrera iteració. La sensació de seguretat era evidentment poc comparable respecte la primera. L'estimació de les tasques va tornar a tenir una mitjana de 3 punts però en aquest cas va ser molt més encertada que en la resta. Com a mesura per intentar solucionar les incidències amb les interrupcions, el que vam fer va ser disminuir el nombre de pomodoros disponibles que tenia un treballador al dia (de 14 a 12), donant-li així un marge més ampli per distribuir la seva feina en cas que calgui.

Els problemes que ens podíem trobar en el dia a dia no sortien dels normals, és a dir bloquejos per dependències i altres interrupcions que requerien ser resoltes amb urgència. Amb aquestes reunions diàries tot l'equip era conscient dels problemes que podien sorgir i va ser notable l'efecte que va tenir en ells ja que el seu treball va anar sent sprint rere sprint més autònom i no depenien de la resta. En els moments on normalment haguessin interromput la feina d'un company per preguntar un dubte, al ser conscients de que aquest estava en mig d'un pomodoro, concertaven un altre pomodoro d'ajuda en el qual resolien tots aquests dubtes en una breu trucada.

Immediatament després d'això i de tornar a calcular la seva jornada, continuaven amb la feina assignada sense cap mena de problema.

Al final d'aquest experiment, vam rebre unes opinions que acabaven sent les mateixes per tots els membres de l'equip. Aquestes, que inclourem en les conclusions, no impliquen cap impossibilitat d'utilitzar aquesta metodologia ni la veuen com a un pas enrere en la millora de l'equip. El *feedback* va ser generalment positiu, recalcant aspectes com una millor organització del temps individual i un estil de treball més tranquil, fet que tots agraïen molt.

7.9 Resultats

Durant tot l'experiment, i gràcies a l'ús de l'aplicació i la col·laboració de l'equip, hem pogut recollir un gran nombre d'informació relacionada amb tot el treball que l'equip ha realitzar en aquestes 4 setmanes.

Un dels objectius era comparar el temps que aquest equip trigava a realitzar les tasques abans i el que ha trigat durant l'ús de la metodologia híbrida. Per això hem comparat els temps extrets de l'eina que s'utilitzava anteriorment respecte als que hem tret de la nova.

	Scrum	Nova metodologia
Usuari 1	2 dies	3 pomodoros
Usuari 2	1 dia	2 pomodoros
Usuari 3	1 dia	3 pomodoros
Usuari 4	2 dies	4 pomodoros

Figura 28: Comparació de mitjanes del temps necessari per a finalitzar una tasca (4 setmanes de feina)

Com podem veure en la *Figura 28*, la diferència de temps és enorme entre les dues metodologies. Això pot semblar evident a primera vista, ja que limitem l'ús de pomodoros per tasca a 5, obligant a que durant la planificació es fragmentin molt més aquestes tasques. Pot semblar que aquesta comparació no sigui significativa ja que hi ha aquest element que implica subdividir les tasques prèviament. Això però, ofereix un canvi molt important en el dia a dia dels treballadors. Quan una persona es veu dins el mateix problema durant un llarg termini de temps, la necessitat de sortir-ne augmenta exponencialment, provocant una caiguda del rendiment d'aquesta. En el nostre cas però, aquest canvi és constant, a vegades fins i tot hi ha més de 3 o 4 canvis per dia, donant així una major sensació d'alliberació de les responsabilitats.

Per tant, podem dir que en aquest aspecte hem aconseguit una certa millora que, tot i que potser no és fàcil de veure a nivells de tasques i temps, sí que ho és psicològicament, i aquest és un factor que moltes vegades no es té en compte en moltes empreses, el benestar dels seus treballadors.

A continuació, també volíem veure com afectava aquesta metodologia en l'organització del temps i les tasques dels membres de l'equip. Una bona forma de visualitzar-ho, és comparar el percentatge de tasques acabades al final dels sprints. Amb això podrem dir si les planificacions que hem realitzat al principi dels cicles són prou bones i donen estimacions realistes.

	Scrum	Nova metodologia
Usuari 1	70%	85%
Usuari 2	60%	70%
Usuari 3	80%	75%
Usuari 4	66%	90%

Figura 29: Comparació del percentatge de finalització de les tasques al final d'un sprint (4 setmanes de feina)

En la *Figura 29*, observem un clar augment d'aquest percentatge de tasques finalitzades en acabar un sprint. Això demostra una bona organització i una visió més realista del temps dels treballadors i com han d'emprar-lo. Com aquests nombres ens donen una mitjana, cal explicar que surten d'una progressió durant aquestes 4 setmanes d'experiment. En la primera setmana, degut a les estimacions errònies i a la poca experiència amb la gestió d'interrupcions, aquests percentatges eren bastant més baixos. Amb el temps, han anat augmentat ràpidament i hi ha hagut casos on fins i tot s'han completat totes les tasques assignades a un membre de l'equip. Aquesta informació l'hem extreta dels *Burndown Charts* que l'antiga eina ens permet visualitzar i del gestor de tasques de la nova.

Finalment, centrant-nos només en la part de la nova metodologia, volem comprovar els resultats que fan referència al balanç de pomodoros. Per avaluar aquest aspecte, hem generat un seguit de gràfiques que contenen el balanç diari de pomodoros de cadascun dels membres de l'equip, on veurem l'evolució a través d'aquest experiment. Amb això volem comprovar si hi ha dificultats a l'hora d'aprendre de la metodologia o si els resultats de rendiment en aquest aspecte no han sigut els desitjats. Aquestes dades han sigut recol·lectades, com hem dit en el plantejament de l'experiment, a banda de l'aplicació, i se'n poden consultar alguns exemples en l'annex corresponent (A.3).

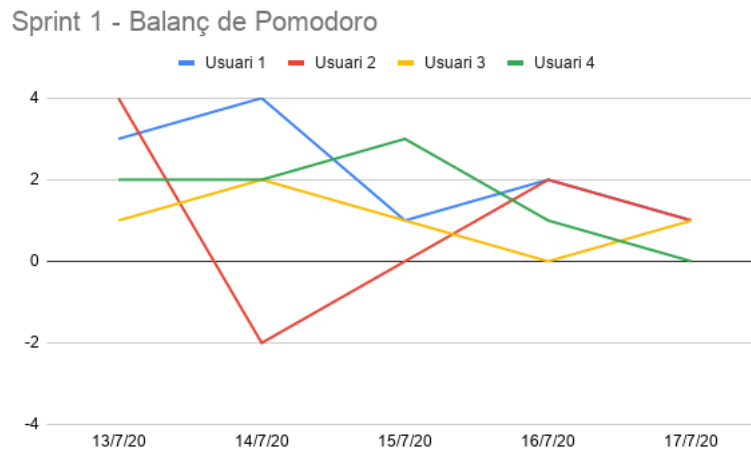


Figura 30: Balanç de pomodoros de l'equip - Sprint 1

Durant el primer sprint, com observem a la *Figura 30*, la majoria d'estimacions són inferiors al que realment la tasca requereix. Aquesta situació, ja d'esperar, ens indica que l'equip encara no s'ha adaptat correctament a la metodologia i que necessiten una millora en les estimacions i un millor control de les interrupcions. També cal notar que els valors més grans es situen al principi del cicle, i van decreixent al final d'aquest. Un possible significat és que després de tornar del cap de setmana, la reconexió amb el treball és més difícil, però un cop començada tot va millorant fins que s'acaba.

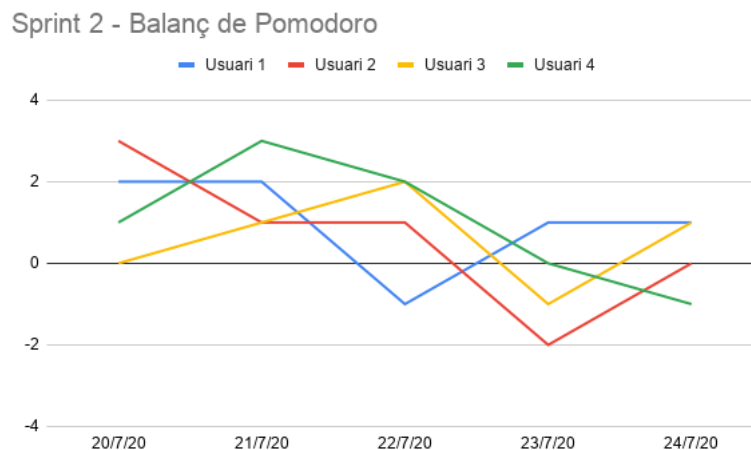


Figura 31: Balanç de pomodoros de l'equip - Sprint 2

En aquest segon sprint (*Figura 31*), observem que la gràfica comença a prendre més resultats negatius tot i que segueix havent alguns de positius. Això ens indica que després de veure els resultats de la primera setmana, les estimacions han començat a ser més altes per culpa d'una por a no complir-les. Tot i això, el problema de les interrupcions externes ha provocat molt soroll en aquestes gràfiques incrementant els resultats per sobre dels esperats.

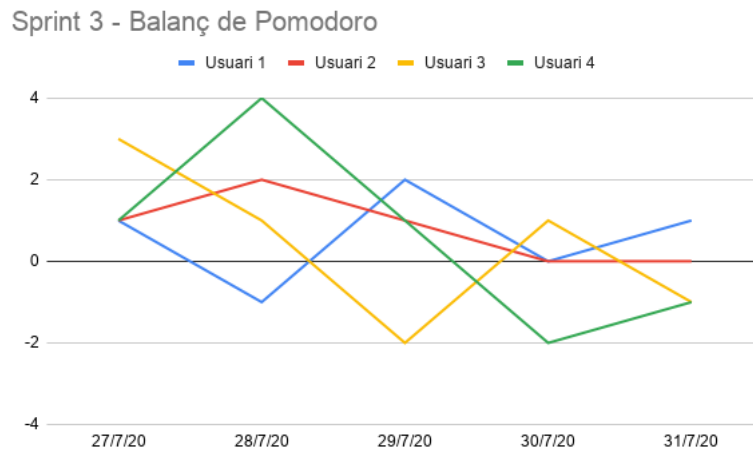


Figura 32: Balanç de pomodoros de l'equip - Sprint 3

Després d'un parell de setmanes seguint la metodologia, veiem que en la majoria de casos la tendència comença a apropar-se més al 0 (*Figura 32*). A més, els resultats són més dispersos, amb dies negatius i positius quasi per igual. Amb això comprovem que segueix la tendència a sobreestimar les tasques però també hi trobem un augment del rendiment que ens aporta valors negatius en alguns d'aquests dies.

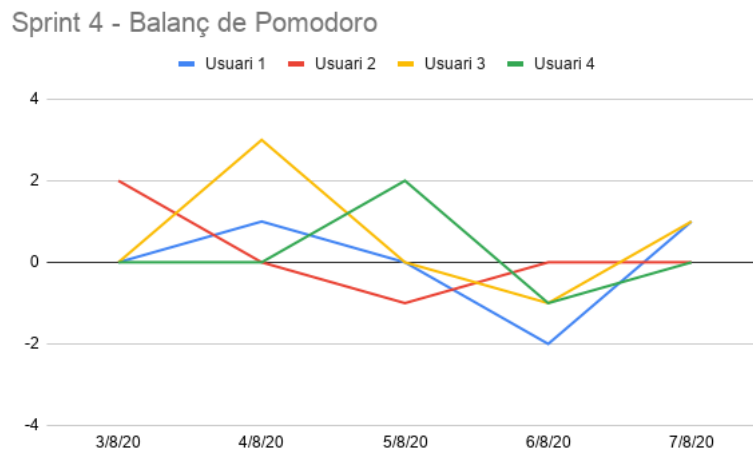


Figura 33: Balanç de pomodoros de l'equip - Sprint 4

Finalment, en aquest últim cicle (*Figura 33*) ja veiem una clara millora tant en les estimacions com en la gestió del temps individual. El fet d'haver reduït la disponibilitat en 2 pomodoros menys, ha aportat més tranquil·litat i possibilitat de gestionar millor el dia a dia. Observem una clara tendència al 0 i només es despunta aquesta gràfica en dos punts durant la primera part de la setmana.

7.10 Conclusions i treball futur

Després de l'anàlisi dels resultats, hem pogut comprovar que hi ha hagut una millora del rendiment significativa en el grup de control sobre el qual hem realitzat aquest experiment. Tant hem pogut veure aquesta millora com una major tranquil·litat en el dia a dia d'aquest, aportant així un millor ambient de treball en el qual els treballadors es sentin a gust.

Sí que és veritat que hem trobat resultats que no seguien al complert la tendència que esperàvem d'un balanç pròxim a 0. Això pot ser degut a que realment ens trobàvem en una fase d'aprenentatge de la metodologia, alhora que comptàvem amb factors externs que dificultaven una mica el fet d'implementar-la correctament, com ara altres equips que no ho feien o la situació de treball remot. Per això, creiem que l'ús d'aquesta metodologia implica que no només l'equip, sinó tothom qui n'estigui en contacte (directius, altres sectors de l'empresa) n'ha de ser conscient d'aquesta, reduint al mínim les interrupcions que puguin causar.

En el que respecta a les tasques, la subdivisió d'aquestes ha implicat a vegades que una tasca es vegi més llarga de l'esperat a l'hora de planificar els diferents cicles. Tot i això, creiem que aquesta subdivisió no només ajuda en la part d'organització de l'equip, sinó també en la claredat dels objectius a complir. També ens hem fixat, comparant amb l'anterior metodologia utilitzada (Scrum), que la majoria de tasques no estan del tot ben especificades. La falta de detalls en aquestes porta a una mala estimació, que tindrà conseqüències en els temps de finalització. En el nostre cas això ens ha afectat i per això també hem observat varies desviacions en els balanços dels pomodoros.

Els treballadors han intentat seguir el màxim possible la metodologia, complint amb els requisits que se'ls imposava. Un fet és que sempre hi haurà interrupcions en un equip amb un nombre considerable de treballadors. Per això veiem una necessitat assignar una disponibilitat de treball menor a la real, per tal de donar marge a aquestes situacions i que els treballadors tinguin l'oportunitat de solucionar-les, ja que tampoc seria bo ni correcte l'acumulació indefinida d'aquestes tasques extra.

Creiem que la metodologia encara té alguns trets per polir, com per exemple aquest que acabem d'esmentar o la disminució al màxim de les interrupcions. Però tot i això hem aconseguit uns resultats considerablement notables i en veiem cert potencial. També veiem com a treball futur una millora de l'aplicació utilitzada. Tot i que realment només fos una eina per a mesurar els resultats de l'experiment, no hem trobat gairebé cap eina que pugui complir amb les mateixes especificacions, així que es podria obtenir un bon producte final si se'n optimitzessin algunes de les seves parts i s'afegissin noves funcionalitats.

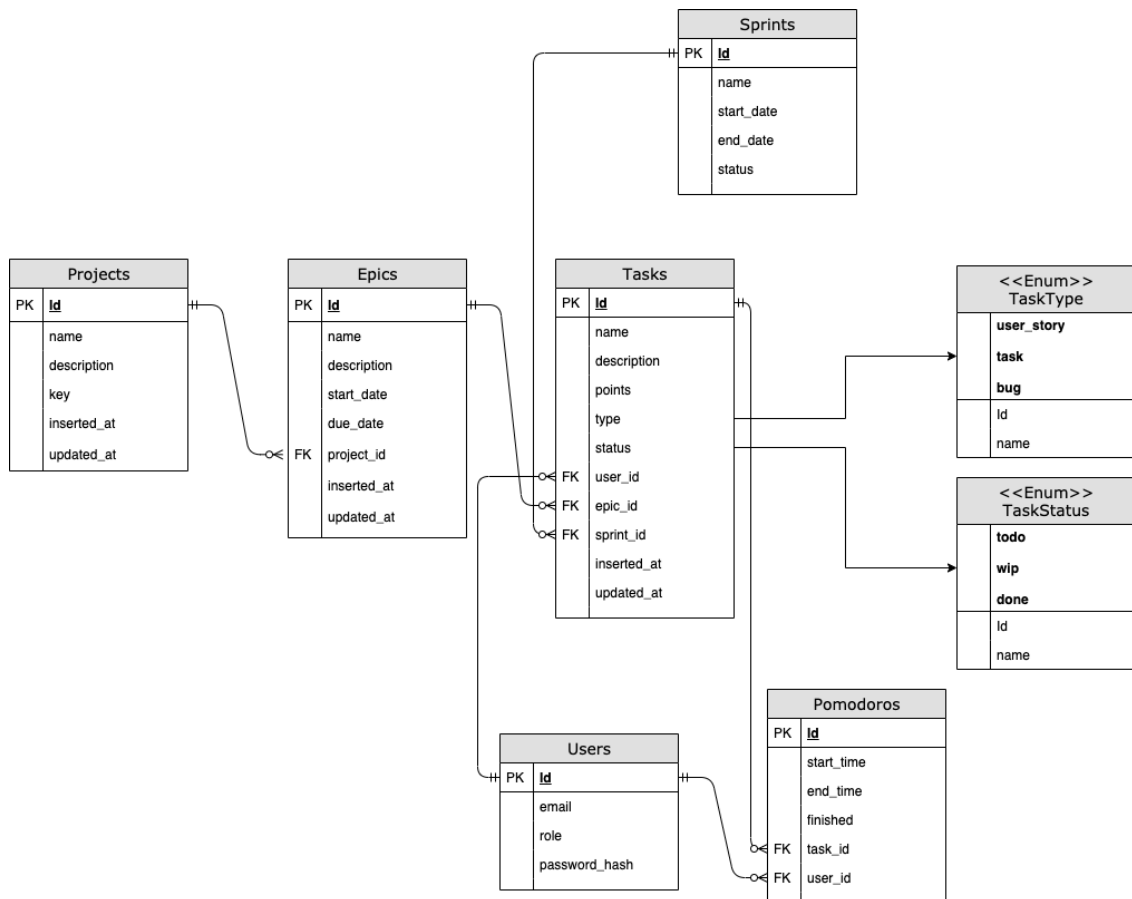
Bibliografia

- [1] Cirillo, F. *The Pomodoro Technique*. San Francisco, Calif: Creative Commons, 2009.
- [2] Ruensuk, M. *An implementation to reduce internal/external interruptions in Agile software development using pomodoro technique*, 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, 2016.
- [3] Beck, K., et al. *The Agile Manifesto*. Agile Alliance.
<http://agilemanifesto.org/agile101/the-agile-manifesto/>, 2001.
- [4] Agile Alliance. *Agile Glossary*. Agile Alliance.
<http://agilemanifesto.org/agile101/agile-glossary/>
- [5] Clockify. *Timeboxing*. Clockify. <https://clockify.me/timeboxing>
- [6] Three five two. *Planning Poker*. Planning Poker.
<https://www.planningpoker.com/>
- [7] Valim, J., et al. *Elixir Language*. Elixir. <https://elixir-lang.org/>
- [8] Valim, J., et al. *Phoenix Framework*. Phoenix Framework.
<https://www.phoenixframework.org/>
- [9] Ericsson AB. *ets*. Erlang – ets. <https://erlang.org/doc/man/ets.html>
- [10] Google. *MVC Architecture*. Chrome developer.
https://developer.chrome.com/apps/app_frameworks
- [11] U.S. General Services Administration. *User Experience Basics*. Usability.
<https://www.usability.gov/what-and-why/user-experience.html>
- [12] Atlassian. *Agile Project Management*. Get started with agile project management. <https://www.atlassian.com/agile/project-management>
- [13] Schultzer, D. *Pow*. Pow auth. <https://powauth.com/>

A Annex

A.1 Model de dades

A continuació deixem una imatge del model de dades complert que hem utilitzat per a l'aplicació:



A.2 Consentiment informat

Model del consentiment informat utilitzat per a dur a terme l'experiment.

CONSENTIMIENTO INFORMADO:

Datos del estudio para el que se otorga el consentimiento

Investigador principal: Xavier Rodríguez Montesinos

Título del proyecto: Análisis de la aplicación de la combinación de metodologías ágiles

Centro: Universitat de Barcelona

Datos del participante

Nombre: — — — —

DNI: — — — —

Persona que proporciona la información y la hoja de consentimiento

Nombre: Xavier Rodríguez Montesinos

1. Declaro que he recibido información sobre el estudio citado.
2. Se me ha entregado una copia de este Consentimiento Informado, fechado y firmado. Se me han explicado las características y el objetivo del estudio, así como los posibles beneficios y riesgos del mismo.
3. He contado con el tiempo y la oportunidad para realizar preguntas y plantear las dudas que poseía. Todas las preguntas fueron respondidas a mi entera satisfacción.
4. Se me ha asegurado que se mantendrá la confidencialidad de mis datos.
5. El consentimiento lo otorgo de manera voluntaria y sé que soy libre de retirarme del estudio en cualquier momento del mismo, por cualquier razón y sin que tenga ningún efecto sobre el trato de mis datos en un futuro.

DOY ☐

NO DOY ☐

Mi consentimiento para la participación en el estudio propuesto

Firmo por duplicado, quedándome con una copia:

Fecha: DD/MM/AAAA

Firma del participante:

“Hago constar que he explicado las características y el objetivo del estudio y sus riesgos y beneficios potenciales a la persona cuyo nombre aparece escrito más arriba. Esta persona otorga su consentimiento por medio de su firma fechada en este documento”.

Fecha: DD/MM/AAAA

Firma del Investigador que proporciona la información y la hoja de consentimiento:

A.3 Informes diaris de treball

Durant la realització de l'experiment, es guardaven diàriament documents de report sobre les tasques realitzades per cadascun dels membres de l'equip. En total tenim 20×4 documents. Donarem seguidament un parell exemples per veure com s'organitzaven i treballaven durant aquests dies. Les tasques i els noms que apareixen en aquests informes han estat canviat per tal de preservar l'anonimat de l'equip i la gent que treballava amb aquest.

ACTIVITIES	pom. estimation	pom. done	pom. cancelled	INTERRUPTIONS
Planning + meeting	2	2	0	Team member
Task cleaning	1	1	0	Short meeting
Architecture with team	1	1	0	Nebulex connection
Implement suggestions	1	4	0	
Umbrella to previous	1	1	0	
EMERGENCIES	pom. estimation	pom. done	pom. cancelled	
Team member 1	1	0	1	
Team member 2	1	1	0	
Discussion	1	1	0	

Figura 34: Planificació de l'usuari 4 pel dia 28/7/2020

ACTIVITIES	pom. estimation	pom. done	pom. cancelled	INTERRUPTIONS
Planning + standup	1	1	0	Algorithm fix
Team planning	1	2	0	VPN issue in Linux not able to connect [fixed]
Pair with team member	2	3	0	
Pair with another team member	2	1	0	
New model for merkel	2	0	0	
Merkel implementation	2	0	0	
Fix CI/CD	1	1	0	
Create migrations for new model	1	2	0	
Review code	1	1	0	
EMERGENCIES	pom. estimation	pom. done	pom. cancelled	
Code removal	1	1	0	

Figura 35: Planificació de l'usuari 2 pel dia 23/7/2020