

Enseñando a diseñar sistemas software para la sostenibilidad: una experiencia docente

Nelly Condori-Fernández^{1,2}, Andrea Herrmann³, Alejandro Catala⁴, Franci Suni⁵
CITIC-Universidade da Coruña (España)¹, Vrije Universiteit Amsterdam (Países Bajos)²,
Heidelberg University (Alemania)³, Departamento de Electrónica e Computación, Centro
Singular de Investigación en Tecnoloxías Intelixentes(CiTIUS), Universidade de Santiago de
Compostela (España)⁴, Universidad de Lima (Perú)⁵
n.condori.fernandez@udc.es, n.condori-fernandez@vu.nl
AndreaHerrmann3@gmx.de, alejandro.catala@usc.es, fsuni@ulima.edu.pe

Resumen

Hoy en día, la sostenibilidad se constituye en uno de los motores claves de la innovación. Así mismo, las empresas de software vienen también demandando nuevas habilidades para continuar su presencia en un mercado muy competitivo. Con el fin de poder atender dichas demandas, una escuela de invierno en modo online fue organizada para complementar la formación brindada en las universidades a los futuros profesionales de ingeniería informática. La escuela de invierno contó con la participación de profesores invitados que cubren 3 módulos: creatividad y software sostenible, diseño de sistemas software persuasivos, e interfaces de usuario interactivas e inteligentes. Los estudiantes trabajaron en equipos centrados en la solución de problemas de un caso real. En el presente artículo se describe el contenido y planificación de cada uno de estos módulos. Además un conjunto de lecciones aprendidas, que son el resultado de analizar los entregables proveídos por los estudiantes y respuestas a un cuestionario de calidad de los cursos impartidos, son también discutidas.

Abstract

Today, sustainability is one of the key drivers of innovation. Likewise, software companies are also demanding new skills to continue their presence in a very competitive market. In order to be able to meet these demands, an online winter school was organized to complement the training provided in universities to future computer engineering professionals. The invited professors at the winter school covered 3 modules: Creativity and sustainable software, design of persuasive software systems, and interactive and intelligent user interfaces. The students worked in teams focused on solving problems of a real case. This article describes the content and planning of each of these modules. In addition, a set of lessons learned, which are the result of analyzing the deliverables provided by the students and responses to a

quality questionnaire of the courses taught, are also discussed.

Palabras clave

Curso en línea basada en equipos, diseño de software sostenible, aprendizaje activo

1. Introducción

Hoy en día, la sostenibilidad se constituye en uno de los motores claves de la innovación. Asimismo, las empresas de software vienen también demandando nuevas habilidades para continuar su presencia en un mercado muy competitivo. Como parte de una escuela de invierno sobre sistemas software persuasivos (SEPS), los estudiantes deben aprender las habilidades necesarias para el desarrollo de software sostenible. Para lograr tal objetivo, SEPS cubre diferentes temáticas organizadas en tres módulos, los cuales facilitaran a los participantes aplicar métodos de creatividad [5] y de diseño de software sostenible [4,10], y herramientas para prototipar interfaces conversacionales a un caso propuesto [2]. De esta manera, los participantes adquieren habilidades en un nivel superior de la taxonomía de Bloom. El caso propuesto, seleccionado cuidadosamente, es utilizado en diferentes cursos planteados con diferentes objetivos de aprendizaje pero complementarios.

Es importante mencionar que debido a la pandemia, SEPS, originalmente planeada como un evento presencial, tuvo que realizarse en modo online. Esto no solo fue un desafío, sino también una oportunidad para aplicar métodos de aprendizaje activo y tener nuevas experiencias como docente y como participante. La colaboración en línea (trabajo en equipos) y el aprendizaje basado en casos es una buena oportunidad de preparación, ya que los participantes adquirirán diferentes habilidades o valorarán aquellos nuevos conocimientos adquiridos que les permitan afrontar mejores trabajos futuros en proyectos de software distribuidos globalmente [7].

En la siguiente sección presentamos los trabajos relacionados respecto a los métodos de enseñanza utilizados. La sección 3 describe la organización de las unidades de aprendizaje de la escuela de invierno de la SEPS. La Sección 4 presenta un análisis de los resultados de aprendizaje, y los retos junto con las lecciones aprendidas son discutidos en la sección 5. El artículo concluye con la sección 6.

2. Trabajos relacionados

En esta sección describimos los trabajos relacionados respecto a las técnicas de enseñanza/aprendizaje utilizadas en los cursos de SEPS.

2.1. Clase invertida

En este método de enseñanza, el estudio de la teoría del curso se realiza en casa y los ejercicios se hacen en el aula. De acuerdo con un reciente estudio de mapeo [14], este método ha sido aplicado exitosamente en diferentes áreas de ingeniería de software (diseño de software, pruebas, calidad de software). También existen experiencias de usar los principios de clase invertida en la enseñanza de programación [6,13], o base de datos [9]. En general, autores de estas experiencias docentes reportan resultados positivos, indicando que el método de clase invertida es más eficiente que el modelo tradicional de lectura y ejercicios. Por ejemplo, Bin Meng [9] concluye que el método puede mejorar la capacidad de aprendizaje autónomo, motivar el interés por aprender y mejorar la comprensión del contenido del curso. Mientras que Ruiz de Miras et al. [13] verificaron mejoras en cuanto a la participación de los estudiantes en debates y un incremento significativo en la asistencia a sesiones de prácticas.

Teniendo en cuenta todas estas ventajas ofrecidas por el método, nosotros decidimos aplicarlo, pero a diferencia de estos trabajos, en un entorno completamente online y con un contenido mucho más específico relacionado a requisitos y diseño de software sostenible. A la fecha existen pocos estudios de aplicar el método en modo virtual [9,15].

2.2. Aprendizaje basado en casos

Este método utiliza casos (reales o hipotéticos) para explotar el aprendizaje teórico del curso. Al estudiante se le dan casos para resolver, lo cual permite que el estudiante aplique los conceptos teóricos aprendidos en clase. Por ejemplo, Ahmad et al. [1] buscan potenciar los conocimientos de seguridad de la información mediante un caso hipotético donde una empresa sufre un robo de propiedad intelectual. Chutisowan et al. [3] proponen combinar el aprendizaje basado en casos y el aprendizaje basado en problemas para el análisis de datos usando programación SQL. A diferencia de estos trabajos, Karsen et al.

[7] combinan dos métodos: aprendizaje basado en casos y clase invertida para evaluar la efectividad de la participación de los estudiantes. En todos estos trabajos se confirma el impacto significativo de aplicar el método en el aprendizaje y la participación de los estudiantes.

2.3. Cursos basados en equipos

Aunque existen diversos artículos académicos relacionados al trabajo en equipo, las empresas de software continúan demandando a las universidades que necesitan profesionales con mejores habilidades para trabajar en entornos basados en equipos [12]. En realidad, a pesar de que en la mayoría de cursos de informática se podría aplicar el trabajo en equipo, muchas veces no es diseñado, ni evaluado adecuadamente, problema que se acentúa aún más cuando el curso es llevado a cabo en modo online.

En Ingeniería de Software (IS), los *Hackathons* y los *Capstone* típicamente se basan en fomentar habilidades de trabajo en equipo, y enfatizar otras habilidades blandas tales como el pensamiento crítico, la gestión de conflictos, la resolución de problemas, entre otras. En este artículo, nosotros enfatizamos el uso de los *hackathons*. Su definición e implementación fueron estudiadas recientemente en una revisión de la literatura [11]. El resultado de dicha revisión fue que el *hackathon* es un término algo vago que representa un evento de corta duración dedicado a resolver algún desafío, por lo general con alguna conexión con la industria o algo específico en un entorno colaborativo. De acuerdo con Porras et al [11], los resultados de aprendizaje más importantes para los participantes de un *hackathon* son: a) habilidades de colaboración, b) creatividad o habilidades de innovación, y c) alguna habilidad específica del contexto (por ejemplo, habilidades en diseño o programación). También se observó que los *hackathons* no son muy utilizados como parte de la educación superior. Los *hackathons* se ven más como una plataforma de aprendizaje informal. En general, los informes de *hackathons* documentan los retos abordados, y no tanto los resultados de aprendizaje.

3. Metodología

3.1. Organización de SEPS

La primera edición de la escuela de invierno sobre sistemas persuasivos para la sostenibilidad (SEPS) ha sido organizada por dos universidades. Debido a la pandemia Covid-19, SEPS tuvo que realizarse en modo virtual (Agosto 2020), originalmente planificado para ser llevado a cabo en modo presencial.

Tal como se observa en la Figura 1, la escuela de invierno ofrece diferentes temáticas organizadas en 3 módulos: Creatividad y Sostenibilidad (mod1), Sis-

temas Persuasivos (mod2), e Interfaces de Usuario (mod3). Es importante mencionar, que los estudiantes tuvieron la flexibilidad de elegir los módulos que les era factible asistir. Esta facilidad se dio principalmente por haber realizado la escuela de invierno en modo online. Además con el fin de facilitar una mejor de retroalimentación en tiempo real, el número de asistentes fue limitado. Para tal, se contó con una fase previa de pre-inscripción, vía formulario web. A continuación detallamos los objetivos, contenido específico, método de enseñanza, número de asistentes y duración de las sesiones de cada temática.

Creatividad.- Este curso fue impartido completamente en modo online con una duración de dos horas sobre creatividad en general y creatividad en IS. Se contó con 50 asistentes. El objetivo del curso fue que los asistentes pudieran aplicar técnicas de creatividad para identificar características y requisitos de un sistema software (chatbot).

En cuanto al contenido del curso, el docente impartió definiciones tanto de IS, como de creatividad, donde los participantes podían proponer, vía chat, características de creatividad. La IS significa aplicar técnicas de ingeniería, utilizar modelos con el fin de optimizar la utilización de recursos, gestionar la complejidad de los procesos de desarrollo, etc. En IS se necesita creatividad en todas las fases del proyecto, no solo para encontrar características innovadoras, o proponer el diseño arquitectural de un producto software, sino también para resolver problemas técnicos que surgen al desarrollar un sistema de software que satisfaga los requisitos. Por consiguiente, con el fin de fomentar acciones e ideas creativas, algunas técnicas de creatividad que puedan ser utilizadas en IS fueron también presentadas. Posteriormente se presentó la descripción de un caso relacionado a la incorporación de nuevos estudiantes a la universidad, y se realizó el análisis de fortalezas, oportunidades, debilidades, y amenazas (FODA), junto con todos los estudiantes, sobre dicho caso. Aunque el análisis de FODA no es una técnica de creatividad, sino una técnica para la planificación estratégica de las empresas, consideramos que puede ayudar a desencadenar nuevas ideas, por ejemplo requisitos innovadores. Finalmente se introdujo el método 6-3-5, y fue también utilizado en el mismo caso propuesto, donde cada equipo tenía que proponer ideas para construir un chatbot que ayude a la incorporación de nuevos estudiantes a la universidad.

Software y sostenibilidad.- Este curso considera la sostenibilidad como una parte integral del software y tiene como objetivo definir un método que ayude a informar las decisiones de diseño impulsadas por la sostenibilidad. Este curso fue organizado en cuatro partes: (i) familiarización con el tema de sostenibilidad, (ii) introducción al kit de herramientas

SAF (Sustainability Assessment Framework) y presentación de ejemplos de mapas de decisión, (iii) aplicación de SAF al caso seleccionado y (iv) presentación de resultados.

A diferencia del curso sobre creatividad, este curso fue impartido utilizando el método de clase invertida para las dos primeras partes. Esto nos facilitó aprovechar mejor la sesión online en la semana de SEPS (Miércoles) para las dos últimas partes (iii y iv). Los estudiantes tenían que revisar el material preparado (video grabado y 1 artículo de lectura) para antes de la clase síncrona.

La sesión online tuvo una duración de dos horas, mientras que la sesión de preparación por parte del estudiante se estimó aproximadamente tres horas. Los organizadores de SEPS proyectaron el video como parte del programa (Lunes). No obstante, los estudiantes podían revisar dicho material en cualquier momento siempre que sea antes de la sesión online.

Interfaces conversacionales.- Este módulo se centra en introducir desde un punto de vista práctico y aplicado lo que son las interfaces de usuario conversacionales. De tal forma, los objetivos fueron: Comprender los conceptos clave asociados a este tipo de interfaces, que incluyen los llamados chatbots y asistentes virtuales; Aplicar dichos conceptos en el diseño y construcción de chatbots, particularizado al manejo de RASA [2] como herramienta de referencia de código abierto; Diseñar un chatbot de universidad considerando aspectos relativos a la sostenibilidad. Para cubrir el primer y el segundo objetivo, se utilizó una aproximación basada en clase expositiva en la que se introdujeron y se ejemplificaron los conceptos mediante múltiples vídeos (aprox. 20 min cada uno), siguiendo un modelo de clase invertida. Al finalizar los vídeos, se incorporaron cuestiones de tipo test de selección múltiple sobre el contenido. Los resultados se utilizaron al principio de la siguiente sesión, síncrona, para soportar una actividad de repaso y retroalimentación colectiva. Dicha sesión, en vivo, se condujo a modo de sesión interactiva en la que la primera parte de la misma se recapituló cómo crear un chatbot con RASA, aprovechando la demostración



Figura 1. Módulos del programa SEPS y resultados de aprendizaje (círculos en azul).

para resolver cuestiones; la segunda parte de la sesión se dedicó a introducir el problema asociado al tercer objetivo e iniciar el trabajo en grupo sobre diseño e implementación. Se instó a los estudiantes a completar la actividad durante la semana en forma de Hackathon online.

Dado que el *aprendizaje basado en casos* es uno de los métodos aplicados en los cursos de estos dos primeros módulos, es importante aclarar que para la formación de equipos recomendamos trabajar con los mismos integrantes de equipo, siempre que fuera posible, a lo largo de la escuela de invierno. Sin embargo, a pesar de haber dado dicha recomendación, fuimos flexibles en cuanto al tamaño. Por ejemplo, en el curso “Creatividad para IS” fue necesario formar equipos de 6 personas; mientras que para el curso “Software y Sostenibilidad” los equipos variaron entre 3 a 5 personas.

Sistemas persuasivos.- El objetivo del curso es doble: (i) entender los principios y características generales de un sistema persuasivo, y (ii) aplicar el análisis de contexto persuasivo del modelo PSD (Persuasive System Design) sobre una aplicación móvil. Este módulo contó con 32 participantes. Se aplicó el modelo estándar de clase invertida, donde los estudiantes tuvieron que leer dos artículos científicos: el primero relacionado al modelo PSD [10] y el segundo sobre la aplicación del modelo en un caso de estudio [6]. La ejecución de esta actividad permitió a los estudiantes lograr el primer objetivo de aprendizaje. Luego, durante la sesión online de 2 horas (aprendizaje síncrono), primero el docente resolvió las preguntas formuladas por los estudiantes respecto al material leído. Y Luego propuso una actividad para que los estudiantes construyan la matriz de cambios/resultados a una aplicación móvil que ayude a manejar alguna situación del covid-19.

En la siguiente sección presentamos los métodos y herramientas que fueron impartidos como parte del contenido de los cursos de la escuela de invierno. La aplicación de estos métodos y herramientas en un caso específico permitió que los estudiantes pudieran aprender a incluir aspectos de sostenibilidad en el diseño de sistemas persuasivos.

3.2. Métodos y herramientas

Método 6-3-5

El método 6-3-5 [5] consiste en formar equipos de 6 personas. Cada persona debe proponer 3 ideas en una hoja de un documento Google. Cada equipo dispone de tan solo 5 minutos para escribirlas. Para completar la hoja es necesario definir un orden de turno entre los participantes. Así cuando un participante termine, el siguiente podrá ver las ideas y

usarlas como información e inspiración. El objetivo es generar nuevas ideas sin que coincidan con las que ya hay escritas. De esta forma se generan 18 ideas diferentes en cada hoja. Esto se repite en 6 rondas, completando 6 hojas, y dando lugar a un total de 108 ideas por cada equipo.

A partir de ahí se podría complementar con la aplicación de otras técnicas para analizar exhaustivamente las ideas generadas para eliminar ideas duplicadas o aquellas que se consideren no relevante o carecen de claridad. El método 6-3-5 fue elegido por su facilidad de uso y porque también nos permite poder trabajar en modo online.

SAF

SAF es un marco de trabajo que ha sido propuesto para guiar la toma de decisiones desde la perspectiva de un diseñador o arquitecto de software [4]. Uno de los componentes de SAF es el **Mapa de decisiones (MD)**, cuya notación visual se muestra en la Figura 2. Un MD enmarca el *impacto esperado* de una arquitectura de software sobre los intereses (*concerns*¹) relevantes de sostenibilidad.

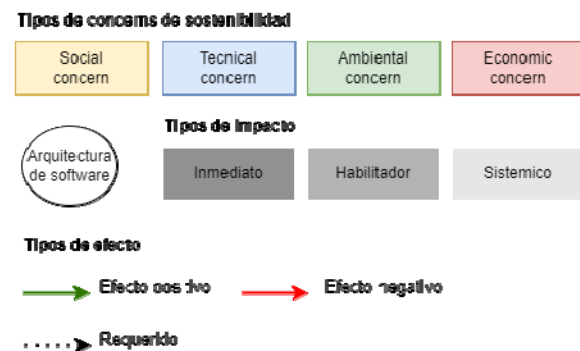


Figura 2. Notación visual de un MD

Según Lago [8], existen tres tipos de impacto esperado: (i) Inmediato: se refiere a cambios inmediatamente observables. Estos se abordan dentro del proyecto de software actual y se espera que sean directamente trazables con entidades de la arquitectura de software. (ii) Habilitador: este tipo de impacto surge del uso del software a lo largo del tiempo. Incluye la oportunidad de consumir más (o menos) recursos, y acortar su vida útil por obsolescencia o sustitución. (iii) Sistémico: se refiere a cambios observables a nivel macro que son persistentes (por ejemplo, cambio de comportamiento humano como consecuencia de usar alguna aplicación software).

En cuanto a los *concerns de sostenibilidad*, SAF considera cuatro tipos: (i) Técnica, aborda el uso de los sistemas software a largo plazo (perdurabilidad),

¹ Un concern captura los requisitos de un sistema.

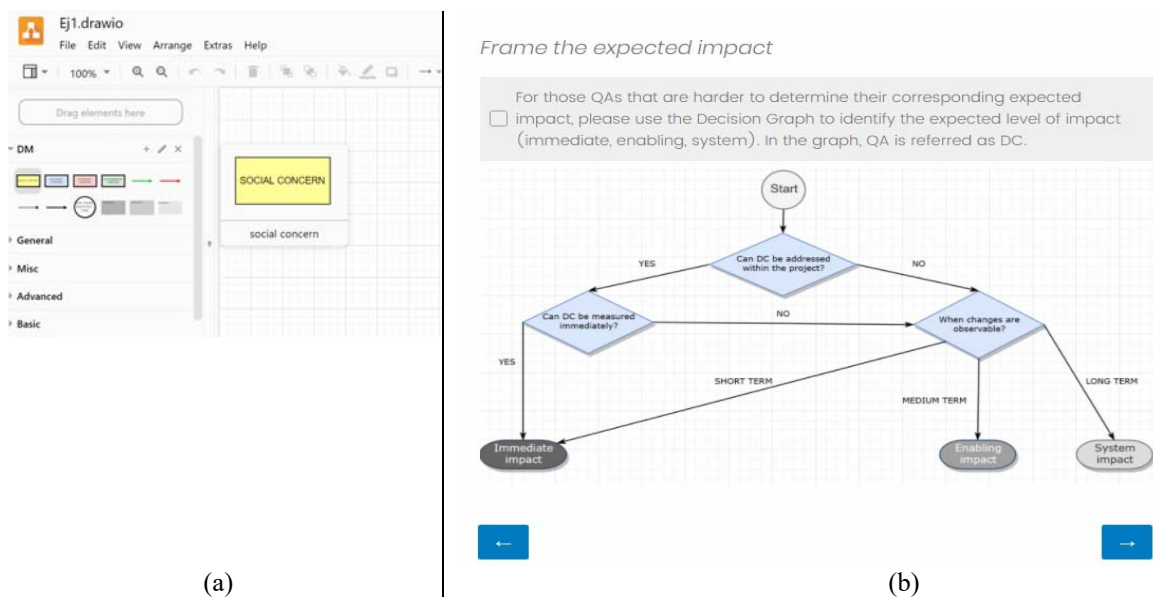


Figura 3. Herramientas del módulo Software y Sostenibilidad: (a) Librería personalizada en Drawio, acorde con la notación visual MD, (b) ejemplo de ítem del checklist para la construcción de un MD.

así como su adecuada evolución en un entorno de ejecución que cambia continuamente.

(ii) Económica, enfoca en preservar el capital y el valor económico. (iii) Social, enfoca en apoyar a las futuras generaciones a tener igual o mayor acceso a los recursos sociales, persiguiendo la equidad generacional. (iv) Ambiental, apunta a mejorar el bienestar humano mientras se protegen los recursos naturales. Las relaciones entre *concerns* se denominan *Efectos*. Existen dos tipos de efectos: positivos y negativos (ver Figura 2).

Un segundo componente de SAF es el **Modelo SQ** que consiste de un conjunto de Atributos de Calidad (ACs) clasificados en los cuatro tipos de *concerns* de sostenibilidad, p.ej. seguridad clasificado como un *concern* de sostenibilidad técnica, y eficiencia energética clasificado como *concern* de tipo ambiental.

Dado que los AC son dependientes entre sí, dicha dependencia puede ser de dos tipos: (i) interdimensional, si relaciona un par de ACs clasificados en diferentes tipos de *concerns*, e (ii) intradimensional, si existe una dependencia entre dos AC clasificados en un mismo tipo de *concern* (por ejemplo, en la sostenibilidad técnica, la seguridad puede depender de la confiabilidad). El modelo SQ, como instrumento de SAF, brinda apoyo para identificar (i) *concerns* relevantes de sostenibilidad, particularmente aquellas relacionadas con los AC; y (ii) tipos de efecto mediante las dependencias entre AC.

Para facilitar la creación de un MD (resultado de aprendizaje), los siguientes **instrumentos de SAF²** fueron proporcionados: lista de atributos de calidad

definidos y clasificados en uno más tipos de sostenibilidad; conjunto de matrices de dependencia, que representan las dependencias inter-dimensionales del modelo SQ; grafo de decisión, el cual fue propuesto para facilitar la correcta identificación de los tipos de impacto esperado; librería personalizada para la herramienta de edición Draw.io², el cual fue utilizado para dibujar los MD; y un checklist (o lista de verificación) basado en web, el cual fue creado para guiar la construcción de un MD.

Por otro lado, con el fin de facilitar el trabajo en equipos de todos los cursos de SEPS, utilizamos la herramienta Zoom y Google drive para tener un solo espacio con los materiales preparados para SEPS, el cual fue organizado por días de semana.

4. Resultados de aprendizaje

En SEPS esperamos que nuestros estudiantes aprendan a utilizar técnicas de creatividad para facilitar la identificación de relevantes *concerns* de sostenibilidad, modelarlos en "mapas de decisión" y reflexionar sobre el impacto de la sostenibilidad de las decisiones tomadas durante proyectos interdisciplinarios con respecto a los tipos de sostenibilidad (técnica, económica, social y ambiental). El tipo de sistemas que enfocamos en SEPS son los sistemas persuasivos, y el tipo de interfaces que consideramos son las interfaces conversacionales.

Por lo tanto, tal como se observa en la Figura 1, los resultados (productos entregables) están relacionados con: 1) Ideas generadas, 2) Mapa de decisiones, y 3) prototipo de interfaz conversacional para un chatbot propuesto para ayudar a la incorporación de nuevos estudiantes a una universidad. Con el fin de poder

² <https://github.com/S2-group/SAF-Toolkit>

determinar en qué medida los estudiantes aprovecharon de los cursos de SEPS, primero evaluamos los entregables producidos por los respectivos equipos de trabajo. Los criterios considerados se presentan en el Cuadro 1. Los porcentajes de generación de ideas, AC y CS relevantes fueron calculados en función al número total de ítems propuestos por cada grupo, la relevancia de cada respuesta fue determinada por el docente del respectivo curso; mientras que el porcentaje de participación fue calculado en función al tamaño del equipo (6 personas).

- **Ideas generadas:** Analizando las plantillas utilizadas por los 8 equipos de estudiantes, siguiendo el método 6-3-5, observamos en el Cuadro 2 que en general los miembros de los grupos G1, G3, G5 participaron activamente en la generación de ideas. Siendo los grupos G3 y G5 quienes propusieron ideas interesantes a considerar en la implementación de un chatbot. Los niveles de satisfacción fueron definidas en una escala ordinal de 4 puntos: muy satisfecho (☺☺), satisfecho (☺), poco satisfecho (☹), nada satisfecho (☹☹). La satisfacción fue definida en función a las potenciales características que pudieron haber sido observadas a partir de las ideas generadas.

De los grupos con baja participación por parte de sus integrantes (G2, G4, y G7), observamos que los grupos G4 y G7 aún pudieron proponer ideas interesantes. Sin embargo, el grupo G2 no pudo realizar la actividad satisfactoriamente, por más que intentó 1 miembro del equipo proponer alguna idea. La participación de G8 fue nula.

- **Mapa de decisiones:** Analizando los MD producidos por los 5 equipos de trabajo, y teniendo en cuenta el porcentaje de identificación de concerns de diseño relevantes, observamos: Los equipos en general identificaron un buen porcentaje de concerns relacionados con los AC que contribuyen algún tipo de sostenibilidad, mientras que experimentaron mayores dificultades para identificar concerns de sostenibilidad (CS). Solo 2 equipos identificaron este tipo de concerns (G2_{ss} y G3_{ss}). Por ejemplo en G3_{ss} consideraron: *Pérdidas de puestos de trabajo, falta de expertos para mantener el chatbot, y capacidad multilinguaje* como CS de tipo social. En cuanto a la precisión en la identificación del tipo de impacto (inmediato, habilitador, sistémico), observamos que en general los tipos de concerns fueron enmarcados en el tipo de impacto de manera adecuada. Sólo G5_{ss} presentó más errores (ver Cuadro 3).
- **Prototipo:** Todos los equipos pudieron formular preguntas y respuestas con una variedad mode-

rada. Sin embargo, solo 2 de los 5 equipos aportaron un prototipo de chatbot.

Entregable(RA)	Criterio
Ideas generadas con el metodo635	% de participación % de generación de ideas Satisfacción del trabajo en equipo
Mapa de decisiones	% de AC relevantes % de CS relevantes Precisión del tipo de impacto
Prototipo de interfaz conversacionales	Variedad de preguntas y respuestas formuladas Prototipado en implementación

Cuadro 1. Criterios de evaluación por RA

	% de participación	% de generación de ideas	Total de ideas	Satisfacción
G1 _c	83%	93%	15	☺
G2 _c	17%	33%	1	☹☹
G3 _c	100%	94%	17	☺☺
G4 _c	50%	100%	9	☺
G5 _c	83%	100%	15	☺☺
G7 _c	33%	100%	6	☺

Cuadro 2. Performance y satisfacción del trabajo en equipo en generación de ideas (primera ronda)

	% de AC relevantes	% de CS relevantes	Precisión (Tipo de Impacto)
G1 _{ss}	100% (18/18)	NI	3 errores
G2 _{ss}	88% (8/9)	67%(4/6)	1 error
G3 _{ss}	90%(10/11)	75% (3/4)	3 errores
G4 _{ss}	100% (13/13)	NI	4 errores
G5 _{ss}	88%(16/18)	(0%) 0/1	9 errores

Cuadro 3. Performance del trabajo en equipo para la creación de MD usando el marco SAF

5. Discusión

El análisis de los respectivos RA, y la evaluación de los estudiantes sobre la implementación de los cursos, nos ha permitido identificar un conjunto de retos que afrontamos al aplicar las técnicas de enseñanza. Dichos retos son presentados en el Cuadro 4. En segundo lugar, listamos algunas lecciones aprendidas (LA) desde la perspectiva del docente:

LA1: El material de aprendizaje debe incorporar actividades interactivas. Las preguntas tipo tests planteadas como parte de la preparación del material (clase invertida) y su utilización en el repaso en vivo son muy útiles para ver la participación y resolver dudas planteadas por los estudiantes. En el material se les incidió que debían responder estos tests para poder hacer la discusión.

	Reto	Discusión
Aprendizaje basado en casos	<i>R1: Comunicación fluida y enfocada</i>	Debido a que la técnica se basa en la discusión, reflexión y capacidad de los participantes para transmitir sus puntos de vista, la buena comunicación es esencial. Sin embargo, diferentes problemas de comunicación pueden surgir. Por ejemplo, algunos participantes pueden buscar dominar la discusión; o que en ciertos equipos resulte difícil que los participantes hablen entre ellos; o que les sea difícil relacionar sus opiniones al contenido teórico del curso, lo que pudiera dar lugar a una discusión desenfocada. Por lo tanto, los profesores deben estar atentos en detectar este tipo de problemas para ayudar a promover la discusión mediante la formulación de preguntas.
	<i>R2: Selección adecuada de un caso representativo para diferentes cursos</i>	La preparación de casos para la enseñanza es una actividad intensa, especialmente cuando planificamos utilizar el mismo caso a través de diferentes cursos y objetivos didácticos.
Clase invertida	<i>R3: Preparación de material comprensible</i>	La naturaleza autodidacta de las aulas invertidas, especialmente el componente que se lleva a cabo fuera del aula, presenta un reto en la preparación de material didáctico que sea fácil de comprender por todos los asistentes. Esto se hace aún más difícil cuando se cuenta con una diversidad de especialidades y experiencia por parte de los participantes. Además el material pre-grabado y las mismas sesiones en línea grabadas dan lugar a que puedan ser revisadas múltiples veces. Esto podría ocasionar utilizar mayor tiempo si el contenido del material no es claro.
Hackathon en línea	<i>R4: Verificación del efectivo trabajo en equipo</i>	Para el docente, la invisibilidad de las actividades de los estudiantes dificulta verificar si el trabajo fue realizado realmente en equipo.
	<i>R5: Comunicación interna fluida</i>	Mantener una comunicación fluida durante las sesiones online entre estudiantes y docentes es ya un reto, en un hackathon en línea lo es aún más sobre toda para mantener una comunicación interna entre los miembros del equipo.

Cuadro 4. Retos para cada una de las técnicas de enseñanza/aprendizaje utilizada en SEPS (online)

LA2: Contexto de los participantes. En los cursos que requerían algún tipo de software, por ejemplo “interfaces conversacionales”, no pudo ser instalado por limitaciones técnicas que tuvieron ciertos participantes (sistema operativo, etc.). Conviene proveer una imagen ligera de Docker o una máquina virtual, la cual debería estar disponible con suficiente antelación para que puedan descargar en función del ancho de banda de su país.

LA3: El caso debe ser presentado desde diferentes perspectivas. El caso tiene un alcance transversal, pero el enfoque en cada curso es particular a la materia. Por lo tanto, cada docente debiera hacer énfasis de los puntos que considere importantes del caso para el cumplimiento de sus objetivos.

LA4: Incentivos. Como por ejemplo un certificado de aprovechamiento podría haber incrementado la entrega/compleción de tareas, especialmente para aquella que fueron planteadas como parte del hackathon llevado a cabo en la semana siguiente.

LA5: Las plataformas disponibles necesitan ser mejoradas para proveer una mejor retroalimentación. Se utilizaron breakout rooms de Zoom para organizar las discusiones y trabajo en equipo. Aunque el docente circula entre ellas para dar retroalimentación, aclarar cuestiones, etc., no es del todo ágil porque requiere ir cambiando de sala, sin criterios que prioricen las necesidades específicas de los estudiantes, lo

que sí puedes gestionar cuando los grupos son presenciales.

LA6: Considerar mecanismos que faciliten el seguimiento del trabajo durante un hackathon online. En un Hackathon online, la responsabilidad recae en los estudiantes y dependerá de la buena voluntad y entusiasmo de los integrantes de los equipos. Consideramos que deberían establecerse mecanismos de entrega parcial, y preferiblemente establecer horas concretas donde tengan que trabajar y que el docente pueda también acceder y estar disponible, de forma que sea síncrono para el docente. Tras el hackathon sería necesario tener una sesión global, que permita a cada equipo presentar lo realizado al resto, de manera que haya mayor compromiso y aprendizaje.

6. Conclusiones

El presente artículo reporta nuestra experiencia de formar a futuros profesionales interesados en el desarrollo de sistemas software que contribuyen al desarrollo sostenible de nuestra sociedad. Para tal fin, se organizó una escuela de invierno, llamada SEPS, el cual cubrió diferentes temáticas.

Con el fin de cumplir los objetivos de aprendizaje formulados para cada curso presentamos también los métodos, técnicas y herramientas que fueron utilizadas por los participantes a lo largo de SEPS, tales como FODA, 6-3-5, SAF y RASA. Para su enseñan-

za, aplicamos técnicas de aprendizaje activo (clase invertida, aprendiza basado en casos, hackathon) que promovieron principalmente el trabajo en equipo. Al analizar los diferentes entregables producidos por los equipos de trabajo, observamos que la aplicación de técnicas de creatividad y SAF ayudaron a crear mapas de decisión con concerns de diseño relevantes para el caso seleccionado. Sin embargo, solo algunos equipos llegaron a entregar el prototipo de interfaces conversacionales. Como resultado de nuestra experiencia docente, varios retos de haber aplicados las técnicas de aprendizaje activo en modo online y una serie de lecciones aprendidas fueron discutidos. Finalmente, consideramos que la presente experiencia docente podría ser adaptada para que las temáticas abordadas en SEPS pudieran ser impartidas en ciertas asignaturas del grado de informática (Ingeniería de Requisitos, Proyectos de Software, Sistemas de Información, Interacción Persona-Ordenador).

Agradecimientos

Este trabajo ha recibido parcialmente financiación del Programa Erasmus Mundus Joint Master Degree SE4GD-619839, así como de la Consellería de Educación, Universidade e Formación Profesional (acreditación 2019-2022 ED431G-2019/04 y grupo competitivo ED431C2022/19) y el Fondo Europeo de Desarrollo Regional, que reconoce al CiTIUS como centro de investigación del sistema gallego de universidades.

Referencias

- [1] Ahmad Atif, Maynard Sean, Motahhir Sameen, Anderson Ashley. Case-based learning in the management practice of information security: an innovative pedagogical instrument. *Personal and Ubiquitous Computing* 25(5), 853–877 (2021)
- [2] Bocklisch Tom, Faulkner Joey, Pawlowski Nick, y Nichol Alan. (2017). Rasa: Open Source Language Understanding and Dialogue Management. ArXiv, abs/1712.05181.
- [3] Chutisowan Kanoksom, Trinantarat Prekarn, Ratnarangsank Kongnat, Jundang Nattapong, Suwatcharakulthorn Jirawut. Combination of problem-based learning and case-based learning in sql programming for data analysis. *6th International STEM Education Conference* (2021).
- [4] Condori-Fernandez Nelly, Lago Patricia, Luaces Miguel, y Places Angeles. 2020. An Action Research for Improving the Sustainability Assessment Framework Instruments. *Sustainability journal*, 12, no. 4: 1682.
- [5] Drews Gunter y Hillebrand Norbert (2007) Lexikon der Projektmanagementmethoden, München, p. 105.
- [6] Kappinen Pasi, Oinas-Kukkonen Harri, Alahai-vala Tuomas, Jokelainen Terhi, Keranen Anna Maria, Salonurmi Tuire, Savo-lainen Markku (2016). Persuasive user experiences of a health Behavior Change Support System: A 12-month study for prevention of metabolic syndrome. *International. International Journal of Medical Informatics* (2016).
- [7] Karsen Marisa, Kristin Desi, Pangestu Harijanto. Evaluation on student engagement of online case-based learning and flipped learning during covid 19 pandemic. *International Conference on Information Management and Technology*. pp. 271–276 (2021)
- [8] Lago Patricia. Architecture Design Decision Maps for Software Sustainability. In *Proc of the International Conference on Software Engineering, Software Engineering in Society Track*, Canada, 29–31 May 2019
- [9] Meng Bin. Practice of flipped classroom teaching mode of computer course – taking access database as an example. *4th Int Conference on Information Systems and Computer Aided Education*. ACM, USA (2021)
- [10] Oinas-Kukkonen Harri. A foundation for the study of behavior change support systems. *Pers Ubiquit Comput* 17, 1223–1235 (2013).
- [11] Porras Jari, Knutas Antti, Ikonen Jouni, Happonen Ari, Khakurel Jayden, Herala Antti (2019). Code camps and hackathons in education - literature review and lessons learned. *HICSS 2019*, Hawaii, USA, pp. 7750-7759.
- [12] Riebe Linda, Girardi Aantonia, Whitsed Craig. A Systematic Literature Review of Teamwork Pedagogy in Higher Education. *Small Group Research*, Vol. 47, Iss. 6, pp. 619-664. (2016)
- [13] Ruiz de Miras Juan, Balsas-Almagro José Ramon, Garcia-Fernandez Ángel-Luis. Using flipped classroom and peer instruction methodologies to improve introductory computer programming courses. *Comput. Appl. Eng. Educ.*(Jul 2021)
- [14] Veras Nécio, Rocha Lincoln, y Viana Windson. Flipped Classroom in Software Engineering. A Systematic Mapping Study. In *Proc of the 34th Brazilian Symposium on Software Engineering (SBES 20)*. 720–729.
- [15] Yildiz Durak Hatice: Flipped classroom model applications in computing courses: Peer-assisted groups, collaborative group and individual learning. *Comput. Appl. Eng. Educ.* (2022)