



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“DESARROLLO DE UNA APLICACIÓN DE RECURSOS EDUCACIONALES ABIERTOS (OERs) DE PRUEBAS DE SOFTWARE EN DEPARTAMENTOS DE DESARROLLO DE SOFTWARE EN GUAYAQUIL”

INFORME DE MATERIA INTEGRADORA

Previo a la obtención del Título de:

INGENIERO/A EN COMPUTACIÓN

KATTYA ROXANA DESIDERIO SÁNCHEZ

STEPHANY RUBÍ QUIMBA HERRERA

GUAYAQUIL – ECUADOR

AÑO: 2017

AGRADECIMIENTOS

Primero quiero agradecerle a Dios por guiarme durante toda mi carrera y por permitirme llegar hasta aquí. A mi madre Blanca Herrera, quien con todo su esfuerzo logró darme todo lo que necesitaba para lograr ser una profesional, junto con la ayuda de mi padre Carlos Herrera y mi tía Irina Herrera, que es una segunda madre para mí, todos ellos fueron incondicionales para mi formación profesional y humana. Y en especial a mi madre Blanca Esmeralda Huacón, quien me supo aconsejar y siempre confió en mí.

Y, por último, pero no menos importante a nuestra mentora la Ing. Mónica Villavicencio, quien nos dio la oportunidad de trabajar con ella y demostrar de lo que somos capaces y nos brindó su apoyo incondicional. Al Lic. Luis Eduardo Mendoza, quien siempre nos prestó su ayuda y nos motivó a cumplir nuestro objetivo.

Stephany Quimba

En primer lugar, quiero agradecer a Dios y a la virgencita de Guadalupe por darme la luz y fuerza en todo momento a lo largo de mi vida universitaria. A mi madre Norma Sánchez Yagual por brindarme siempre su apoyo incondicional, a mi abuelita que está en el cielo por ser mi ángel que me cuida siempre.

Quiero agradecer a tres personas, maestros en estos últimos años de carrera universitaria. La primera es la PhD. Mónica Villavicencio por bríndame la confianza de ser su ayudante de cátedra y luego de investigación. A mi tutor, Luis Caiza Pérez por haber sido más que mi tutor convirtiéndose en mi amigo, por apoyarme, confiar en mí y en mi trabajo, siempre resaltando mis aptitudes para que no cayera en mis propias desconfianzas. Por último y no menos importante al PhD. Luis Eduardo Mendoza por siempre estar atento a cualquier duda que se me presentará y darnos aliento para continuar. Muchas gracias.

Katty Desiderio

DEDICATORIA

Este proyecto se lo dedico a Dios, mi madre Blanca Herrera, mi padre Carlos Herrera y mi tía Irina Herrera, quienes me guiaron en todo momento y me enseñaron valores y principios para conseguir todo lo que me proponga. En memoria de mi madre Blanca Esmeralda Huacón, quien me dio su amor incondicional, consejos, ánimos y nunca perdió la fe en mí.

Stephany Quimba

El presente proyecto se lo dedico a Dios y a mi madre Norma Sánchez. Mis rocas para continuar cada día de mi vida, en cualquier proyecto que tenga demostrando lo que realmente soy como profesional y como persona.

Katty Desiderio

TRIBUNAL DE EVALUACIÓN

Mónica Villavicencio C., Ph.D.

PROFESOR EVALUADOR

Luis Eduardo Mendoza M., Ph.D.

PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me(nos) corresponde exclusivamente; y doy(damos) mi(nuestro) consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Stephany Rubí Quimba Herrera

Katty Roxana Desiderio Sánchez

RESUMEN

El presente trabajo describe el desarrollo de una aplicación de Recursos Educativos Abiertos (OERs) de pruebas de software dirigido a personas que desarrollan software y se muestran interesados en aprender o reforzar conocimientos con tiempo limitado. Los OERs son materiales de enseñanza-aprendizaje que se pueden utilizar y distribuir gratuitamente.

Como antecedentes, se realizó un estudio a empresas guayaquileñas, donde se detectaron varios problemas en el desarrollo de software tales como: retrasos en la entrega, entrega de producto con errores, entre otros. Existen varias causas que provocaron dichos problemas mencionados anteriormente, entre ellas nos llamó la atención la falta de pruebas de software durante el desarrollo.

Hoy en día, los profesionales buscan recursos en línea para fortalecer sus conocimientos, pero estos suelen ser muy extensos, incompletos y no corresponden al área de interés que desean aprender. Por lo tanto, se tomó la decisión de suplir este problema, desarrollando tres OERs de pruebas de software: caja negra, integración y caja blanca.

Posteriormente se les describirá cómo se llevó a cabo este proyecto. El primer capítulo detalla la importancia de las pruebas de software, los principales problemas en empresas guayaquileñas, sobre OERs de Ingeniería de Software y su importancia. El segundo capítulo, fue dividida en tres fases: primero, cómo se llevó a cabo el diseño de cada OER; segundo, cómo se estructuró cada uno en el LMS y tercero, cómo se planificó el monitoreo a los voluntarios de cada OER. En el tercer capítulo, se detallan los resultados de cada uno de los OERs y se describen los problemas y observaciones encontradas. Y, finalmente, las conclusiones y recomendaciones; entre ellas, se concluyó que el proyecto llenó las expectativas de los profesionales voluntarios, ya que los OERs son precisos y simplificados respecto al tema de enseñanza y los videos son de fácil comprensión.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACIÓN	iv
DECLARACIÓN EXPRESA	v
RESUMEN.....	vi
ÍNDICE GENERAL.....	vii
CAPÍTULO 1	9
1. LA IMPORTANCIA DE LAS PRUEBAS DE SOFTWARE	9
1.1 Principales problemas en los departamentos de software de Guayaquil	10
1.2 Recursos Educativos Abiertos (OERs)	10
1.2.1 Antecedentes	10
1.2.2 Impacto de los OERs en la educación continua	11
1.3 OERs para Ingeniería de Software	13
CAPÍTULO 2.....	15
2. METODOLOGÍA DE TRABAJO	15
2.1 Diseño y Estructura de los OERs	17
2.2 Selección del LMS	17
2.3 Estructuración de los OERs en el LMS	20
2.4 Monitoreo	20
CAPÍTULO 3.....	21
3. RESULTADOS	21
3.1 Caja negra	22
3.2 Integración	22
3.3 Caja blanca	23
3.4 Evaluación de los OERs	24
3.5 Problemas y Observaciones	25

CONCLUSIONES	26
RECOMENDACIONES.....	27
BIBLIOGRAFÍA.....	28
ANEXOS.....	32
Anexo 1: Diseño Instruccional de Caja negra	32
Anexo 2: Diseño Instruccional de Integración.....	34
Anexo 3: Diseño Instruccional de Caja blanca.....	35
Anexo 4: Taller de Caja negra	36
Anexo 4: Taller de Integración	38
Anexo 5: Taller de Caja blanca	43

CAPÍTULO 1

1. LA IMPORTANCIA DE LAS PRUEBAS DE SOFTWARE

El presente capítulo muestra una introducción al problema que deseamos resolver. Se resume en que los programadores de software de la ciudad de Guayaquil no cuentan con recursos educacionales abiertos completos sobre pruebas de software y cómo repercute en la entrega de un producto de software de calidad. Conociendo más sobre el problema, trataremos sobre la estrategia de crear, diseñar, estructurar y poner en práctica recursos educacionales gratuitos en cualquier área de la educación, su impacto y cómo se puede extender hasta a la etapa de probar el producto de software.

El ciclo de vida del software comprende de varias etapas. Entre ellas, se encuentran las pruebas de software, las cuales permiten validar y verificar si los requerimientos cumplen los objetivos para los cuales fueron creados. Pero, sobre todo, para detectar los errores a fin de que éstos puedan ser corregidos para obtener finalmente un producto robusto.

Existen pruebas de caja negra y caja blanca. La primera se encarga de validar entradas y salidas del sistema; es decir, que exista carencia de ambigüedad y consistencia en los requerimientos. La segunda se encarga de realizar un seguimiento al código fuente para comprobar si se cumplen todas sus instrucciones, bloques o caminos, a fin de medir la cobertura del programa.

Es muy importante realizar pruebas de software, ya que sin ellas no se detectan las fallas, las cuales provocan en el futuro fallas mayores, que harán incurrir a las empresas en gastos muy altos.

1.1 Principales problemas en los departamentos de software de Guayaquil

Un grupo de investigación en el área de software de ESPOL, como parte del proyecto “Selección de Recursos Educativos Abiertos (OERs) de ingeniería de software basada en las necesidades de empresas y departamentos de desarrollo de software en la ciudad de Guayaquil” [1], realizó una encuesta a empresas guayaquileñas para identificar los problemas a los cuales se enfrentaron y qué los causó. El 50,6% de las empresas indicaron que una de las causas fue el no priorizar correctamente los requerimientos y un 34,1% indicaron que fue la falta de pruebas de software.

El presente trabajo se enfoca en crear, investigar y aplicar OERs de pruebas de software a profesionales voluntarios que trabajan y disponen de poco tiempo.

Sin embargo, antes de profundizar en los OERs de pruebas de software es importante conocer un poco más acerca de qué son y cuál es su impacto ante la sociedad.

1.2 Recursos Educativos Abiertos (OERs)

1.2.1 Antecedentes

A través de los años, la tecnología ha transformado e innovado la forma de enseñar y aprender, facultando a profesores y alumnos a utilizar la tecnología para su propio beneficio [2]. La evolución en el uso de los recursos educativos de apoyo nos ha permitido observar una inclusión de herramientas sustentadas en tecnología [3].

Las Tecnologías de Información y Comunicación (TIC) permiten distribuir el conocimiento [4], así como gestionar y transformar la información [5]. Por tanto, las TICs se han convertido en herramientas para el diseño de estrategias innovadoras educativas que permitan mejorar el diseño de los ambientes de aprendizaje para potencializar las experiencias de enseñanza-aprendizaje [6].

La UNESCO indica que el acceso universal a la educación es esencial para el desarrollo sostenible de la sociedad, el diálogo intercultural, entre otros aspectos. La apertura en la educación no es

fundamentalmente digital, aunque sí ha ido incrementando cada vez más con el pasar de los años [8]. Esta iniciativa se está ampliando al material docente, a los denominados Recursos Educativos Abiertos (Open Educational Resources - OER por sus siglas en inglés.) [7]. Los OERs son materiales de enseñanza y aprendizaje que han sido publicados para que su utilización, adaptación y distribución, sea gratuita [25].

Los OERs son utilizados en diversas áreas de conocimiento y su uso trae consigo muchas ventajas, no sólo para la academia, sino también para el sector productivo [15]. Aunque cabe señalar que no existe una definición formal para los OERs ya que las definiciones varían en función del aspecto que se quiere enfatizar bien sea la reutilización o el ámbito de aplicación [7].

1.2.2 Impacto de los OERs en la educación continua

A nivel mundial, los OERs están impactando en la educación superior [9], ya que permiten la apertura y la flexibilidad del proceso de aprendizaje, proporcionando oportunidades adecuadas y adaptadas a las necesidades individuales [10].

Las instituciones educativas son beneficiadas con el empleo y desarrollo de OERs, ya que esto les permite llegar a un número mayor de estudiantes, optimizar el tiempo invertido en la producción de contenidos, vincularse con otras instituciones dentro y fuera del sector educativo para crear redes de colaboración, entre otros beneficios [18]. Para las empresas, el usar OERs en las capacitaciones de sus empleados significa potenciales reducciones de costos y un mejoramiento significativo de las habilidades de su personal y, por ende, posibles incrementos de productividad [19].

Los OERs incluyen cursos y programas curriculares, módulos didácticos, guías de estudiante, libros de texto, artículos de investigación, vídeos, podcasts, herramientas de evaluación, materiales interactivos (como simulaciones), bases de datos, software,

aplicaciones (incluyendo aplicaciones móviles) [12] y cualquier otro material educativo diseñado para uso en la enseñanza y el aprendizaje [13].

Los OERs pueden jugar un papel importante en cualquier sistema de educación en donde la innovación, personalización, alta calidad, reducción de costo y eficiencia, van de la mano. Sin embargo, el saber cómo producir materiales relevantes y de calidad que pueden ser reutilizados y adaptados en diferentes contextos y aprendizajes suele ser un trabajo difícil para sus creadores [10].

Es importante recalcar que los OERs son seleccionados y evaluados bajo criterios tales como: tiempo, costo, propósito, área de conocimiento, idioma, calidad, usabilidad, ambiente colaborativo, entre otros. La evaluación se debe a que son fuentes de información y fortalecimiento de conocimiento. Otros expertos consideran que un OER está definido por cuatro R: reusar, redistribuir, revisar y remezclar [14]. Reusar, permite a las personas usar todo o parte del trabajo para sus propios fines; Redistribuir, para que las personas puedan compartir con otros su trabajo; Revisar, con el fin de que las personas pueden adaptar, modificar y traducir el trabajo compartido y Remezclar, debido a que pueden tomar 2 o más recursos y combinarlos entre ellos.

Por lo tanto, crear, seleccionar o evaluar un OER depende de muchos factores para que éste pueda ser accesible y, sobre todo, sea útil y tenga valor para la sociedad.

1.3 OERs para Ingeniería de Software

El desarrollo de software ha sido considerado tanto un arte como una ciencia [20]. Como ciencia, esta disciplina se fundamenta en la aplicación de prácticas de ingeniería que permiten estimar, medir y evaluar el proceso de desarrollo, de manera repetitiva y controlada. Como arte, el desarrollo de software requiere producir resultados que reflejen ingenio y habilidad para seleccionar, diseñar y construir el producto software que mejor satisfaga los requerimientos de una organización [21].

Existen varios tipos de OERs que se aplican en ciertas etapas del desarrollo de software, pero estos recursos son más orientados a requerimientos, aprendizaje de nuevos lenguajes, prototipado, entre otros. Sin embargo, una etapa muy importante y que posee pocos o muy extensos recursos de aprendizaje son las “Pruebas de Software”. De acuerdo con el proyecto realizado por ESPOL, tal como se mencionó en la sección 1.1, una de las causas por la que muchos profesionales no toman dichos cursos es porque son muy extensos, y el factor tiempo para ellos es muy importante, ya sea por trabajo, estudios o familia, entre otros [1]. O simplemente para la tarea que debe realizar en el trabajo no se ajusta al tiempo de entrega. Es por esa razón que la mayoría de desarrolladores optan por no realizar pruebas de software para cumplir con los tiempos de entrega. Recordemos que los profesionales buscan recursos que faciliten el aprendizaje en poco tiempo para aplicarlo en las tareas asignadas y facilitar su trabajo, logrando ser más eficientes y entregar un trabajo a tiempo.

De acuerdo con lo anteriormente mencionado, nuestra investigación busca crear, estructurar y aplicar OERs de pruebas de software. En vista que el tiempo es un factor problemático para cualquier investigación, más cuando se depende del tiempo que disponen los profesionales voluntarios, se seleccionaron los siguientes tipos de pruebas para la creación de OERs:

OERs 1: Caja negra.

OERs 2: Pruebas de Integración.

OERs 3: Pruebas de Caja Blanca.

Por cada OER se realizó un diseño instruccional para organizar los diferentes recursos como: videos, diapositivas, talleres y evaluaciones.

Luego se escogió BrainCert como Sistema de gestión del aprendizaje (Learning Management System - LMS por sus siglas en inglés) para compartir los OERs y dar seguimiento a los voluntarios.

Con el fin de incentivar y lograr que los voluntarios concluyeran el OER en BrainCert. Adicionalmente, los voluntarios recibieron asesoría y asistencia sobre el uso de la herramienta, aplicación de teoría para talleres prácticos y retroalimentación del resultado de las evaluaciones que se realizaron. Cada actividad y tarea que se planificó tuvo como objetivo impulsar el desarrollo educacional del voluntario para demostrar que al poseer un conocimiento más completo sobre pruebas de software se puede disminuir los problemas de software que puede tener un producto.

CAPÍTULO 2

2. METODOLOGÍA DE TRABAJO

Como se indicó en el capítulo anterior, nuestro estudio se basó en un estudio previo en el que se realizó una selección de OERs basado en las necesidades de empresas y departamentos de desarrollo de software en la ciudad de Guayaquil.

El presente capítulo muestra la metodología de trabajo usada en este proyecto la cual tiene tres fases. La primera, en la que se detalla cómo se llevó a cabo el diseño de cada OER, la segunda en la que se muestra cómo se estructuró cada uno en el LMS y la última en la que se explica cómo se planificó el monitoreo por parte de nosotros a los voluntarios de cada aplicación del OER respectivo, que en capítulo tres se lo detalla.

Al ser un material educativo, debimos seguir lineamientos ya establecidos para asegurar que el desarrollo del mismo siga estándares. Para esto se creó por cada OER un diseño instruccional, el cual fue proporcionado por Jhonny Pincay, Ing. en Computación, Coordinador de Curso Herramientas de Colaboración Digital (HCD).

Se realizó una recopilación de todo el material teórico y práctico existente, seleccionando el material útil y completo, que podía ser sólo visual (diapositivas) o audiovisual (video).

Con respecto al material práctico que se usó para los talleres que realizaron y completaron los voluntarios, se investigó sobre herramientas open source que complementaron el material de pruebas de software proporcionado en el OER, estos fueron: Github, Travis y Coveralls.

Git es un software de control de versiones muy popular entre los desarrolladores. GitHub. Es un servicio de alojamiento de repositorios de software, el cual permite revisar, contribuir, corregir errores de un código fuente facilitando el trabajo en equipo [26].

Travis CI es un sistema distribuido de integración continua que permite conectar cualquier repositorio de GitHub, en donde se definen los test para que la herramienta verifique si la integración se realizó correctamente o caso contrario identificar donde se produjo el error [27].

Coveralls es una herramienta que permite medir el porcentaje de cobertura del código de un programa [28].

Estas herramientas sirvieron de apoyo para poner en práctica los conocimientos adquiridos por medio de los recursos otorgados en LMS. Es muy importante dar a conocer dichas herramientas, ya que, facilitan el trabajo cuando se desarrolla software y son totalmente gratuitas. En caso de querer más beneficios pueden acceder a ellas pagando una tarifa.

El diseño instruccional (ver tabla 1) está compuesto por varios campos, tales como: tema, objetivo, contenido, tiempo y evaluación. El tema, concierne en dos subcampos, el nombre del área a cubrir y las diferentes sub áreas que comprende dicho tema. Objetivo, describe lo que quiere lograr dicho tema. Contenido, se divide en dos subcampos: tipo de contenido y recurso digital; en el primero se indica si el contenido es: conceptual, procedimental o condicional, y en el segundo se indica si es video, lectura o foro. Tiempos, la cantidad que le toma al participante en usar el contenido. Evaluación, se divide en dos subcampos: tipo de actividad y autoevaluación; el primero se enfoca en lo que quiere lograr la actividad y el segundo puede ser formativa o sumativa (las preguntas pueden ser tipo: opción múltiple, verdadero o falso, drag drop, etc). El diseño instruccional permite lograr tener un orden de los temas a cubrir y evidenciar que cada actividad o tarea realizada sea acorde con el objetivo de cada OER.

DISEÑO INSTRUCCIONAL DEL MOOC							
TEMA		OBJETIVO	CONTENIDO		TIEMPO	EVALUACIÓN	
SEMANA/TEMA	LECCIÓN/SUBÁREA	OBJETIVO	TIPO DE CONTENIDO	RECURSO DIGITAL	TIEMPO DE DEDICACIÓN	TIPO DE ACTIVIDAD	AUTOEVALUACIÓN

Tabla 1: Plantilla del diseño instruccional.

2.1 Diseño y Estructura de los OERs

Para cada OER se decidió que debía contener 3 elementos principales: teoría, ejercicios y un taller práctico, y la forma en que se evaluaría sería por medio de un pre-test y un post-test para medir si el participante mejoró el rendimiento después de haberlo utilizado. Y, por último, la evaluación del OER utilizado, el cual cuestiona sobre criterios tales como: calidad de contenido, valor educativo, percepción del usuario, motivación, comentarios, entre otros. Para mayor detalle se puede observar en los anexos al final del documento (Ver Anexos).

Antes de aplicar los OERs a cada representante de las empresas que participaron en el estudio, se realizó una prueba piloto en el curso de Ingeniería de Software II, dictado para la carrera de Ingeniería en Computación. Se realizó esto debido a que tratándose de empresas debíamos corregir errores en redacción por incompletitud, ambigüedad y demás, que pudiesen haber tenido los ejercicios para evitar que los representantes que participen en el estudio no llegaran a completar la tarea/actividad debido a que el OER haya tenido errores.

2.2 Selección del LMS

Hoy en día, la oferta cada vez más frecuente de cursos en línea ha dado paso a la creación de cursos enfocados al desarrollo de software, tales como: videos tutoriales y los MOOC (acrónimo en inglés de Massive Open Online Course) o COMA en español (curso online masivo abierto). MOOCs que abarcan desde la enseñanza de lenguajes de programación hasta las prácticas de ingeniería de software con el propósito de garantizar la calidad del producto desarrollado.

Al ser nuestro objetivo comprobar que el uso de los OERs mejora el rendimiento de la persona que lo usa en el proceso de desarrollo de software y la tendencia

inminente de los MOOCs, optamos por escoger una plataforma en línea para no sólo alojar nuestro material educativo sino también medir el rendimiento de los participantes.

Actualmente existe una gama de MOOCs en el web, a pesar de que es una tecnología relativamente nueva. Poco a poco ha ganado terreno en las aulas de clases de universidades de renombre alrededor del mundo. En el 2012, se consolidaron las principales plataformas que ofrecen este tipo de cursos (entre otras edX, que incluye cursos de las universidades de Harvard, Berkeley, y el MIT, Coursera, Canvas, Khan Academy, Udacity, o la plataforma en español MiríadaX).

Después de evaluar plataformas de e-learning populares como: Moodle, edX, Xerte, entre otros, elegimos la plataforma bajo ciertos criterios (ver tabla 2), el cual encajó mejor para nuestro estudio, es por ello que se decidió usar BrainCert (<https://www.braincert.com/>) como LMS.

PLATAFORMAS / CRITERIOS	MOODLE	EDX	XERTE	BRAIN CERT
Facilidad de uso.	X	X		X
Navegación jerárquica.				X
Proporciona un camino claro en la navegación. (Contenido del curso: Test, Talleres, Diapositivas, etc)				X
Acceso gratuito como profesor.	X			X
Acceso gratuito como estudiante.	X			X
Permite colaboración entre participantes.	X	X	X	X
Desarrollo de cursos gratuitos.	X		X	X
Funciones administrativas.	X	X	X	X
Gestión de contenido.	X	X	X	X
Reportes personalizados.	X	X	X	
Visualización del flujo de trabajo de capacitación de los participantes.	X			X
Plataforma de pruebas en línea.	X	X		X
Usuarios ilimitados en plan gratuito.	X			X
Plataforma web.	X	X		X
Plataforma móvil.	X	X		X
Idiomas soportados.	Multi-Idioma	Multi-Idioma	Multi-Idioma	Sólo Ingles

Tabla 2: Tabla comparativa sobre plataformas LMS.

BrainCert es un LMS de formación en línea basada en la nube. Permite a sus usuarios crear y vender cursos ilimitados en línea. Permite crear cursos online y brinda herramientas de personalización. Pueden acceder al curso sin límites de estudiantes; sólo requiere de registrarse con un email. Actualmente es utilizada por: Concordia University, University of Arkansas, Oregon State University, entre otros.

2.3 Estructuración de los OERs en el LMS

Una vez seleccionado a BrainCert como LMS procedimos a llevar a cabo la estructuración de cada OER de manera independiente con su respectivo contenido. BrainCert permite crear n cantidad de cursos con una cuenta gratuita, lo que facultó para que nosotros creáramos los tres OER: caja negra, integración y caja blanca de manera independiente. En vista a que para cada curso se creó un diseño instruccional, sólo fue cuestión de entender como BrainCert estructura un curso, para colocar el contenido del OER en LMS y poder invitar a los profesionales voluntarios.

Cada curso que se crea lo permite dividir en capítulos o chapters y subdividir en lecturas o lectures. Dentro de cada lectura se añadió el contenido visual (diapositivas, pdfs, docs.), audiovisual (videos), las evaluaciones (Google Forms y las de BrainCert) y las reglas para completar las lecturas. Cada lectura debía ser completada después que el material mostrado haya sido leído completamente o en caso de un taller realizado y enviado. Los profesionales voluntarios podían navegar entre lecturas completadas una vez que marcaban la casilla de “Mark as Learn” que era el que le permitía proseguir con el orden secuencial establecido del curso.

2.4 Monitoreo

Para el monitoreo en BrainCert, proporciona una visualización del progreso de todos los profesionales voluntarios participantes de cada curso que se crea y asocia el progreso de los participantes con tres colores: gris que significa que esta enrollado en el curso, amarillo si está en progreso y verde si ha finalizado. Esto nos permitió dar seguimiento a los profesionales voluntarios con el fin de incentivarlos y ayudarlos para que cumplan con los tiempos asignados para finalizar el curso.

En el capítulo 3 indicaremos los resultados de la aplicación de los OER en cada curso que se creó en BrainCert.

CAPÍTULO 3

3. RESULTADOS

El capítulo contiene los resultados que se obtuvieron de la aplicación de los tres OERs de pruebas de software en el siguiente orden: caja negra, integración y caja blanca. Los resultados a presentar por cada OER son: Pre y Post test. Luego, detallaremos de forma general los resultados respecto a la evaluación de los OERs. Y, por último, los problemas y observaciones que tuvieron los profesionales voluntarios en todo el proceso del uso de los OERs. En este capítulo se analizan los resultados de cada OER y nos referiremos al mismo como curso.

Primero, dado el conjunto de participantes se los clasificó a los profesionales voluntarios por su status actual; (ver tabla 3), es decir, si es un estudiante (E), si es un profesor (P) o si trabaja fuera de ESPOL (T).

TIPO	
E-T	Estudiante-Trabaja
T	Trabaja
P	Profesor

Tabla 3: Clasificación de los participantes por tipo.

También se los clasificó por el progreso que tienen dentro del curso en la herramienta "BrainCert"; (ver tabla 4) si sólo ingresó, pero no continuó (I), si está en progreso y si aún no completa el curso (P) o si ya finalizó el curso (F).

ESTADO	
I	Ingresado
P	En Progreso
F	Finalizado

Tabla 4: Clasificación de los participantes por estado dentro de la plataforma BrainCert.

A continuación, se presentarán por cada curso los respectivos resultados.

3.1 Caja negra

El curso tuvo un total de 18 participantes de los cuales: 7 se registraron en el curso, pero no continuaron, 3 ingresaron al curso y siguen en progreso, y los 8 restantes terminaron el curso realizando todos los test, talleres y evaluaciones. Entre los participantes, 11 de ellos están estudiando y trabajando a la vez, 2 profesores y 5 sólo trabajan.

De los 8 que terminaron el curso, 5 son estudiantes que trabajan, 1 profesor y 2 que sólo trabajan. Se calculó la mediana por cada tipo de participante en el Pre y Post test, los datos se presentan en la tabla 5:

TIPO	PRE-TEST	POST-TEST
E-T	63	63
P	50	50
T	56,50	63

Tabla 5: Mediana del puntaje sobre 100 del Pre y Post Test clasificado por tipo del curso de Caja Negra.

La tabla 5, muestra que los estudiantes que trabajan se mantuvieron en el Pre y Post test. Mientras que, los profesores obtuvieron un puntaje menor en los test y los que sólo trabajan mejoraron progresivamente del Pre al Post test.

3.2 Integración

El curso tuvo un total de 13 participantes de los cuales: 4 se registraron en el curso, pero no continuaron, 2 ingresaron al curso y continúan en progreso, y los 7 restantes terminaron el curso realizando todos los test, talleres y evaluaciones. Entre los participantes, 6 de ellos están estudiando y trabajando a la vez, 2 profesores y 5 sólo trabajan.

De los 7 que terminaron el curso, 4 son estudiantes que trabajaban, 1 profesor y 2 sólo trabajan. Se calculó la mediana por cada tipo de participante en el Pre y Post test, los datos se presentan en la tabla 6:

TIPO	PRE-TEST	POST-TEST
E-T	50	80
P	40	60
T	40	80

Tabla 6: Mediana del puntaje sobre 100 del Pre y Post Test clasificado por tipo del curso de Integración.

La tabla 6 muestra que de forma general todos mejoraron progresivamente sus puntajes del Pre al Post test. Los estudiantes que trabajan aumentaron el doble en el Post test. Mientras que, los profesores aumentaron en menor proporción en el Post Test y los que sólo trabajan mejoraron progresivamente en el Post Test.

3.3 Caja blanca

El curso tuvo un total de 13 participantes de los cuales: 2 se registraron en el curso, pero no continuaron, 3 ingresaron al curso y continúan en progreso, y los 8 restantes terminaron el curso realizando todos los test, talleres y evaluaciones. Entre los participantes, 3 de ellos están estudiando y trabajando a la vez, 1 profesor y 9 sólo trabajan.

De los 8 que terminaron el curso, 2 son estudiantes que trabajan, 1 profesor y 5 que sólo trabajan. Se calculó la mediana por cada tipo de participante en el Pre y Post test, los datos se presentan en la tabla 7.

TIPO	PRE-TEST	POST-TEST
E-T	77,50	94
P	60	88
T	50	75

Tabla 7: Mediana del puntaje sobre 100 del Pre y Post Test clasificado por tipo del curso de Caja Blanca.

La tabla de resultados muestra que de forma general todos obtuvieron puntajes superiores e iguales a 60. Los estudiantes que trabajan mejoraron en el post test. Mientras que, los profesores aumentaron en el Post Test y los que sólo trabajan mejoraron progresivamente en el Post Test.

3.4 Evaluación de los OERs

Al finalizar cada uno de los cursos, los participantes calificaron los OERs, respecto a cuatro criterios: Calidad del Contenido, Valor educativo, Diseño/Presentación y Motivación, donde se procedió a realizar un boxplot (Ver figura 3.1):

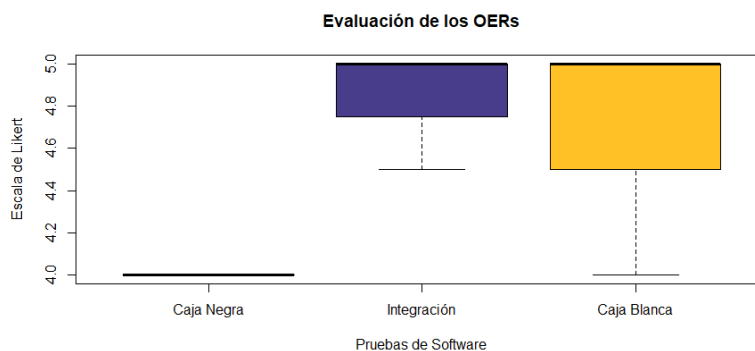


Figura 3.1: Evaluación de los OERs medida con la escala de Likert obtenido de la mediana de todos los criterios

De acuerdo con la figura 3.1, podemos observar que el OER de integración tuvo mayor aceptación con una escala de 4,5 a 5; es decir, los participantes indicaron estar totalmente de acuerdo con la mayoría de las características de cada uno de los criterios. A diferencia de caja negra y caja blanca, que obtuvieron una escala de 4 y de 4 a 5, respectivamente.

Por lo tanto, los OERs fueron de gran aceptación y cumplieron las expectativas de los participantes. Se infiere que una de las razones por las que integración obtuvo mejores resultados porque el curso fue el más corto, se dio un seguimiento diario, asesoría en línea o presencial y se utilizaron otras herramientas en línea tales como: GitHub y Travis.

Adicionalmente, dentro de la evaluación del OER se les pidió observaciones o comentarios respecto al mismo. Y como respuesta obtuvimos:

OER Caja Negra:

“Los videos son de fácil comprensión. Las diapositivas muestran lo necesario, poca información, pero más que suficiente para poder comprender el tema.”

OER Integración:

“Los videos son muy concisos y simplificados, las diapositivas están bien resumidas y los ejemplos son muy buenos.”

OER Caja Blanca:

“Los videos son muy interactivos y particularmente me fueron muy útiles para aprender rápidamente los conceptos claves. Las diapositivas podrían incluir más ejercicios de criterios de caminos, ramas...”

3.5 Problemas y Observaciones

Los problemas que surgieron a los participantes en el momento de utilizar el recurso de caja negra fueron más de usabilidad con la plataforma. Por ejemplo, una de ellas es que no podían continuar a la siguiente lectura y eso era a causa de que no daban clic en “Mark as learned”. La mayoría de participantes tuvieron inconvenientes al manejar Google forms dentro de la plataforma, puesto que creían que no se podía continuar respondiendo, y el problema en realidad es que la herramienta no permitía hacer desplazamiento con alguna barra sino con el tab; por lo tanto, en las instrucciones se indicó que tomaran en cuenta eso.

CONCLUSIONES

En pruebas de integración y caja blanca se logró mayor participación debido a que había un seguimiento diario, fue más corto y se usaron más herramientas open source, lo que fue apreciado por los profesionales voluntarios por los diversos beneficios que cada herramienta ofrece.

Los participantes que culminaron los OER en menor tiempo fueron quienes pidieron asesoría de forma presencial o en línea, en donde sus dudas se resolvieron de manera inmediata.

Una de las características más denotada en los comentarios fue que los OERs son muy simplificados y concisos con la información respecto al tema a tratar en el OER convirtiéndolo en un recurso corto y adaptable al tiempo disponible.

Uno de los principales beneficios de los OERs de pruebas de software es que son una combinación de recursos (videos, diapositivas, lecturas, talleres, etc.) logrando proporcionar diversidad de información para que el participante logre comprenderlo desde cualquier perspectiva.

Además, logramos evitar que las personas interesadas en el área de ingeniería de software inviertan tiempo en la búsqueda de recursos de su interés, ya que proporcionamos OERs de pruebas de software con recursos nuevos y existentes.

RECOMENDACIONES

En pruebas de integración y caja blanca se logró mayor participación debido a que había un seguimiento diario, fue más corto y se usaron más herramientas open source, lo que fue apreciado por los profesionales voluntarios por los diversos beneficios que cada herramienta ofrece.

Los participantes que culminaron los OER en menor tiempo fueron quienes pidieron asesoría de forma presencial o en línea, en donde sus dudas se resolvieron de manera inmediata.

Una de las características más denotada en los comentarios fue que los OERs son muy simplificados y concisos con la información respecto al tema a tratar en el OER convirtiéndolo en un recurso corto y adaptable al tiempo disponible.

Uno de los principales beneficios de los OERs de pruebas de software es que son una combinación de recursos (videos, diapositivas, lecturas, talleres, etc.) logrando proporcionar diversidad de información para que el participante logre comprenderlo desde cualquier perspectiva.

Además, logramos evitar que las personas interesadas en el área de ingeniería de software inviertan tiempo en la búsqueda de recursos de su interés, ya que proporcionamos OERs de pruebas de software con recursos nuevos y existentes.

BIBLIOGRAFÍA

- [1] A. Oyola Suárez (2016). "Selección de Recursos Educativos Abiertos (OERs) de ingeniería de software basada en las necesidades de empresas y departamentos de desarrollo de software en la ciudad de Guayaquil".
- [2] J. Sánchez Ramírez, J. Elizalde Salas, C. Juárez Landín, M. Martínez Reyes and A. Soberanes, "Implementación de un recurso educativo abierto para la enseñanza de Matemáticas basado en GeoGebra", Ri.uaemex.mx. Available: <http://ri.uaemex.mx/handle/20.500.11799/49972>.
- [3] M. Ledo, I. Sánchez, G. González and G. Hernández, "Recursos educativos abiertos", Ems.sld.cu. Available: <http://www.ems.sld.cu/index.php/ems/article/view/233/119>.
- [4] Ramírez Montoya, M. S., & Burgos Aguilar, J. V. (2012). "Recursos Educativos Abiertos en Ambientes Enriquecidos con Tecnología". Available: <http://catedra.ruv.itesm.mx/bitstream/987654321/566/8/ebook>.
- [5] Duarte, E. S. (2008). "Las tecnologías de información y comunicación (TIC) desde una perspectiva social" Revista Electrónica Educare, 12, 155-162. Available: <http://www.redalyc.org/pdf/1941/194114584020.pdf>.
- [6] Aguila, J. V. B. (2010). "Distribución de conocimiento y acceso libre a la información con Recursos Educativos Abiertos (REA)". Available: http://www.educoea.org/portal/La_Educacion_Digital/laeducacion_143/articles/reavla dimirburgos.pdf.
- [7] Santos-Hermosa, G., Ferran Ferrer, N., & Abadal, E. (2012). "Recursos educativos abiertos: repositorios y uso". Available: <http://www.accesoabierto.net/sites/accesoabierto.net/files/Santos-Ferran-Abadal-EPI.pdf>.
- [8] L. Havemann, "Open Educational Resources", 2017.

- [9] M. Ramírez Montoya and F. Mortera Gutiérrez, "Implementación y desarrollo del portal académico de Recursos Educativos Abiertos (REA): Knowledge Hub para educación básica", Hdl.handle.net. Available: <http://hdl.handle.net/11285/577828>. [Accessed: 05- Jun- 2017].
- [10] Arimoto, M., Barroca, L., & Barbosa, E. F. (2016). "Developing Open Educational Resources through Learning Design and Agile Practices". Available: <http://www.br-ie.org/pub/index.php/sbie/article/viewFile/6704/4592>.
- [11] Butcher, N. (2015). "A basic guide to open educational resources (OER)". Commonwealth of Learning, Vancouver and UNESCO. Available: <http://unesdoc.unesco.org/images/0021/002158/215804e.pdf>.
- [12] Atkins, D. E., Brown, J. S., & Hammond, A. L. (2007). "A review of the open educational resources (OER) movement: Achievements, challenges, and new opportunities" (pp. 1-84). Creative common. Available: <http://www.hewlett.org/wp-content/uploads/2016/08/ReviewoftheOERMovement.pdf>.
- [13] Lecercle, D. (2011). Giving knowledge for free: "The emergence of open educational resources". Organization for Economic Co-operation and Development. Available: <https://www.oecd.org/edu/ceri/38654317.pdf>.
- [14] J. Hilton III, D. Wiley, J. Stein and A. Johnson, "The four 'R's of openness and ALMS analysis: frameworks for open educational resources", *Open Learning: The Journal of Open and Distance Learning*, vol. 25, no. 1, pp. 37-44, 2010.
- [15] M. Cedillo, M. Peralta, P. Reyes, D. Romero and M. Toledo, "Aplicación de recursos educativos abiertos (REAs) en cinco prácticas educativa con niños mexicanos de 6 a 12 años de edad", Hdl.handle.net. Available: <http://hdl.handle.net/10486/661275>.
- [16] Iniesto, F., & Rodrigo, C. (2013). "Estándares y accesibilidad en el ciclo de creación de OERs mediante herramientas de autor". Available: <http://www.esvial.org/atica2013/>.

- [17] Masie, E. (2002). "Making sense of learning specifications & standards: A decision maker's guide to their adoption". The Masie Center, evaluation.
- [18] "Open educational resources (OERs)", Jisc. Available: <https://www.jisc.ac.uk/guides/open-educational-resources>.
- [19] "University-Industry Collaboration for Software Engineering Teaching - IEEE Xplore Document", ieeexplore.ieee.org. Available: <https://ieeexplore.ieee.org/abstract/document/4709402/>.
- [20] S.Lawrence (2002). "Ingeniería de software. Teoría y práctica." Brasil, Editorial Prentice Hall/Pearson Educación.
- [21] R. Anaya, "Una visión de la enseñanza de la Ingeniería de Software como apoyo al mejoramiento de las empresas de software", Publicaciones.eafit.edu.co. Available: <http://publicaciones.eafit.edu.co/index.php/revista-universidad-eafit/article/view/809>.
- [22] Herrera, Lizka Johany, and J. Lizka. (2003) "Ingeniería De Requerimientos Ingeniería De Software." Available: <http://www.monografias.com/trabajos6/resof/resof.shtml>
- [23] Aguera, J.Vázquez, P.Hernández, M.González, L.Altamirano, Y.Hernández, and R.Contreras. (2011) "Tópicos Selectos para la Enseñanza de la Ingeniería de Software: Introducción a la Ingeniería de Software."
- [24] "Towards the evaluation of open educational resources for learning software engineering - IEEE Xplore Document", ieeexplore.ieee.org. Available: <https://ieeexplore.ieee.org/abstract/document/7833338/>.
- [25] García, C. M. (2015). "Diseño e implementación de cursos abiertos masivos en línea (MOOC): expectativas y consideraciones prácticas". Revista de Educación a Distancia, (39). Available: <http://www.um.es/ead/red/39/mendez.pdf>.
- [26] "Features For collaborative coding - developers work better, together | GitHub", GitHub, 2017. [Online]. Available: <https://github.com/features>.

[27] "Travis CI - Test and Deploy Your Code with Confidence", Travis-ci.org, 2017. [Online]. Available: <https://travis-ci.org/>.

[28] "Coveralls.io - Test Coverage History and Statistics", Coveralls.io, 2017. [Online]. Available: <https://coveralls.io/>.

ANEXOS

Anexo 1: Diseño Instruccional de Caja negra

DISEÑO INSTRUCCIONAL							
TEMA		OBJETIVO	CONTENIDO		TIEMPO	EVALUACIÓN	
SEMANA /TEMA	LECCIÓN/SUBÁREA	OBJETIVO	TIPO DE CONTENIDO	RECURSO DIGITAL	TIEMPO DE DEDICACIÓN	TIPO DE ACTIVIDAD	AUTOEVALUACIÓN
	PRE TEST	Pre evaluación sobre pruebas de caja negra			10m		
Pruebas de Caja Negra	Pruebas funcionales	Introducción a Pruebas funcionales o Caja negra.	Conceptual	video	1:21s	Comprensión	Ninguna
		Descripción, características de pruebas funcionales.	Conceptual	lectura	3m	Comprensión	Formativa - Verdadero o Falso
	Clases de equivalencia	Descripción de las clases de equivalencias y cómo se identifican.	Conceptual	video	3:12s	Comprensión	Formativa - Drag & Drop
		Elaboración de las clases de equivalencia.	Procedimental	lectura	5m	Aplicación	Formativa - Ejercicios de Opciones Múltiples
	Casos de Prueba	Descripción de los casos de prueba.	Conceptual	lectura	3m	Comprensión	Formativa - Verdadero o Falso
		Elaboración de los casos de prueba.	Procedimental	video	3:30s	Aplicación	Formativa - Ejercicios de Opciones Múltiples
	Valores de Frontera	Descripción de los valores de frontera y cómo se identifican.	Conceptual	lectura	3m	Comprensión	Formativa - Drag & Drop
		Clases de equivalencia que no poseen valores de	Conceptual	lectura	3m	Comprensión	Formativa - Ejercicios de Opciones Múltiples
		Identificar las intersecciones de clases de equivalencias para generar casos de prueba.	Conceptual	lectura	3m	Comprensión	Formativa - Verdadero o Falso

Taller	Aplicar todos los conocimientos aprendidos de las lecturas anteriores	Procedimental		30m		Ninguna
POST TEST	Evaluación sobre pruebas de caja negra			20m		
Evaluación del OER	Evaluar el OER			10m		

Anexo 2: Diseño Instruccional de Integración

DISEÑO INSTRUCCIONAL							
TEMA		OBJETIVO	CONTENIDO		TIEMPO	EVALUACIÓN	
SEMANA /TEMA	LECCIÓN/SUBÁREA	OBJETIVO	TIPO DE CONTENIDO	RECURSO DIGITAL	TIEMPO DE DEDICACIÓN	TIPO DE ACTIVIDAD	AUTOEVALUACIÓN
Pruebas de Integración	PRE TEST	Pre evaluación sobre pruebas de integración			15m		
	¿Cómo y por qué usar las pruebas de integración ?	Conocer cómo y por qué usar las pruebas de integración	Conceptual	video	3:58s	Comprensión	Formativa- Drag & Drop
	Tipos de Pruebas	Descripción de los Tipos de Pruebas y ejemplos	Conceptual	video	3:04s	Comprensión	Formativa - Verdadero o Falso
	Herramientas a usar	Descripción y ejemplo de como usar las herramientas Git, Travis y Coveralls	Conceptual	video	24m	Comprensión	Ninguna
	Taller	Aplicar los conocimientos aprendidos en las lecturas anteriores sobre pruebas de integración y usar Travis	Procedimental	Lecturas	30m	Aplicación	Ninguna
	POST TEST	Evaluación sobre pruebas de integración			15m		
	Evaluación del OER	Evaluar el OER			5m		

Anexo 3: Diseño Instruccional de Caja blanca

DISEÑO INSTRUCCIONAL							
TEMA		OBJETIVO	CONTENIDO		TIEMPO	EVALUACIÓN	
SEMANA /TEMA	LECCIÓN/SUBÁREA	OBJETIVO	TIPO DE CONTENIDO	RECURSO DIGITAL	TIEMPO DE DEDICACIÓN	TIPO DE ACTIVIDAD	AUTOEVALUACIÓN
Pruebas de Caja Blanca	PRE TEST	Pre evaluación sobre pruebas de caja blanca			15m		
	Pruebas de caja blanca	Introducción a Pruebas de caja blanca	Conceptual	video	5m	Comprensión	Ninguna
	Tipos de Pruebas	Descripción sobre las Cobertura de camino	Procedimental	video	3m	Aplicación	Formativa - Verdadero o Falso
		Características y ejemplo de Cobertura de sentencia	Procedimental	video	5m	Aplicación	Formativa- Drag & Drop
		Características sobre Cobertura de condición	Procedimental	video	3m	Aplicación	Formativa - Verdadero o Falso
		Características sobre Cobertura rama	Procedimental	video	6m	Aplicación	Formativa - Verdadero o Falso
		Ejemplos sobre Cobertura de sentencias y ramas	Procedimental	Lecturas	3m	Aplicación	Formativa- Drag & Drop
	Herramientas a usar	Descripción y ejemplo de como usar las herramientas Git, Travis y Coveralls	Conceptual	video	24m	Comprensión	Ninguna
	Taller	Ejercicio para aplicar los criterios de cobertura usando las herramientas: Github, travis y coveralls	Procedimental	Lecturas	30m	Aplicación	Ninguna
	POST TEST	Evaluación sobre pruebas de caja blanca			15m		
Evaluación del	Evaluar el OER			5m			

Anexo 4: Taller de Caja negra

APP(Pharmacys)

La distribuidora farmacéutica “Pharmacys” a lo largo de los años se ha caracterizado por brindar a sus clientes un servicio de calidad. La empresa se encuentra ubicada en Guayaquil, Quito, Cuenca y se está extendiendo en otras ciudades del Ecuador.

Con el fin de solucionar problemas del día a día, “Pharmacys” desea contratarlo para desarrollar una aplicación móvil que permita gestionar el envío a domicilio de medicinas u otros productos que se encuentran disponibles en su catálogo.

Para realizar el pedido, el cliente debe de ingresar a la app con su usuario y contraseña, y luego ingresar la siguiente información:

- Producto
- Cantidad
- Forma de pago (Efectivo o Tarjeta de Crédito)

Los pedidos a domicilio no pueden exceder de 3 productos.

La empresa quiere lanzar un piloto de la app en Guayaquil, ciudad en la cual solo tiene habilitada la entrega en tres sectores: Urdesa (Norte), 9 de Octubre desde la Av. Quito a Malecón (Centro) y los alrededores de Riocentro Sur (Sur).

Para el sector Norte, el servicio a domicilio funciona desde las 8 am hasta las 11 pm, mientras que para los otros dos sectores hasta las 9 pm. El monto de los pedidos a domicilio debe ser mayor a \$10 para evitar recargos adicionales por transporte. El recargo por transporte es de \$2 para las entregas al Sur y al Centro, y \$4 para las del Norte. La app le mostrará al cliente: la cantidad a pagar por los productos y el recargo correspondiente.

Se pide lo siguiente:

- 1) Identifique las clases de equivalencia de todas las variables que intervienen en el pedido a domicilio.
- 2) Escriba los casos de prueba necesarios para asegurar que los horarios de atención por sector sean manejados correctamente por la app.

Favor adjuntar sus respuestas en un documento y enviarlo o compartirlo (Google Docs) al correo stephyquim@gmail.com

Anexo 4: Taller de Integración

APP(Pharmacys)

La distribuidora farmacéutica “Pharmacys” desea contratarlo para desarrollar una aplicación móvil que permita gestionar el envío a domicilio de medicinas u otros productos que se encuentra disponibles en su catálogo, y que sus clientes podrán adquirirlos seleccionándolos en la app por medio de un carrito de compras para proceder al envío. Recordemos que, para realizar el pedido, el cliente debe de ingresar a la app con su usuario y contraseña.

“Pharmacys” le solicita desarrollar lo siguiente:

- Permitir crear productos.
- Visualizar productos en un catálogo(lista).
- Agregar productos al carrito de compra.
- Visualizar el detalle del pedido con los productos seleccionados.
- Confirmar el Pedido, para esto, el cliente debe de ingresar el nombre, sector donde vive y la forma de pago (Efectivo o Tarjeta de Crédito).
- Generar el resumen del pago = total a pagar por los productos más recargo.

Los pedidos a domicilio no pueden exceder de 3 productos. Por cada producto, sólo se pueden solicitar hasta un máximo de 5 unidades.

La app será habilitada sólo en la ciudad de Guayaquil, y en tres sectores: Urdesa (sector Norte), 9 de octubre desde la Av. Quito a Malecón (sector Centro) y los alrededores de Riocentro Sur (sector Sur).

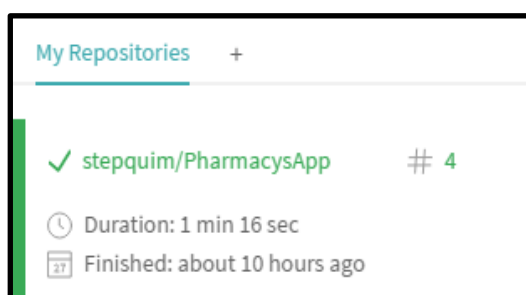
Para el sector Norte, el servicio a domicilio funciona desde las 8 am hasta las 11 pm, mientras que para los otros dos sectores hasta las 9 pm. El monto de los pedidos a domicilio debe ser mayor a \$10 para evitar recargos adicionales por transporte. El recargo por transporte es de \$2 para las entregas al Sur y al Centro, y \$4 para las del Norte.

El líder del proyecto de Pharmacys le entrega a usted el código de fuente de la app a fin de que, el cual debe revisarlo y entenderlo para usted comprenda cómo funciona el sistema.

En el siguiente taller usted debe:

1. Leer detenidamente tabla proporcionada con las pruebas de integración del caso de prueba.
2. Participar en el repositorio “<https://github.com/stepquim/PharmacysApp>”, **crear una rama** con su usuario de github en el repositorio y clonar en su local. **El código esta en java y contiene las tres primeras pruebas de integración que se muestran en la tabla proporcionada.**
3. Codificar el caso de prueba número 4.
4. Crear el caso de prueba 5, este debe garantizar la integración de las funcionalidades, desde que se visualiza los productos en el carrito hasta que se muestra el resumen del pago. Agregarlo a la tabla y Codificar.
5. Ingresar a <https://travis-ci.org/> y hacer sign in con su cuenta de **github** o ir a <https://travis-ci.org/stepquim/PharmacysApp>
6. Para ver cómo se ejecutan las pruebas en travis, usted debería haber realizado el respectivo commit y push, luego podrá visualizarlo de lado izquierdo donde:
 - a. Amarillo: Las pruebas se están ejecutando.
 - b. Verde: Las pruebas fueron exitosas.
 - c. Rojo: Las pruebas fallaron.

Como se muestra en el ejemplo:



7. Sugiere mejoras al código de la app (¿está debidamente documentado?, sigue un estándar? ¿Puede optimizarse?

Pruebas de Integración

Id	Nombre	Funcionalidades/Módulos	Descripción	Datos de prueba	Resultados esperados	Resultados obtenidos
1	Chequear el enlace entre crear producto y ver el producto en catálogo.	F1: Crear producto. F2: Ver producto en el catálogo.	El usuario ingresa al sistema, da clic en el botón <i>Crear producto</i> , donde ingresa el nombre y la cantidad de stock, luego da clic en <i>Guardar</i> . El usuario regresa al menú principal y da clic en <i>Ver catálogo</i> .	producto a crear = Ciprofloxacina cantidad en stock= 50. Descripción:"Medicina" Límite de pedido= 15 precio=11.42	El producto creado aparece en la interfaz de Ver catálogo.	El producto creado aparece en la interfaz de Ver catálogo.
2	Ver el producto en catálogo y visualizar el pedido con los productos agregados al carrito.	F1: Ver catálogo. F2: Visualizar pedido con los productos agregados al carrito.	El usuario ingresa al sistema, da clic en el botón <i>Ver catálogo</i> , luego el usuario selecciona un producto y la cantidad; lo agrega al carrito de compras. Después da clic en <i>visualizar el pedido con los productos agregados al carrito</i> .	productos seleccionados = Redoxon cantidad = 5	El usuario visualiza el pedido con todos los productos agregados al carrito de compras y subtotal del producto agregado.	El usuario visualiza el pedido con todos los productos agregados al carrito de compras y subtotal del producto agregado.

Id	Nombre	Módulos	Descripción	Datos de prueba	Resultados esperados	Resultados obtenidos
3	Chequear entre la funcionalidad de Confirmar pedido y el tipo de pago escogido.	F1: Confirmar pedido. F2: Tipo de pago.	El usuario ingresa al sistema, añade los productos deseados al carrito, después da clic en <i>visualizar el pedido con los productos agregados</i> , luego da clic siguiente y el usuario ingresa nombre, sector y tipo de pago y da clic en <i>Confirmar pedido</i> .	productos seleccionados = Analgan, Buscapina cantidad = 1, 1 pago= efectivo Nombre= "Juan" Sector=Norte	El usuario da clic en Confirmar Pedido y se muestra la información que ingreso y el tipo de pago.	El usuario da clic en Confirmar Pedido y se muestra la información que ingreso y el tipo de pago.
4	Chequear entre la funcionalidad de Confirmar pedido, tipo de pago y el horario escogido.	F1: Confirmar pedido. F2: Tipo de Pago F3: Horario de entrega.	El usuario ingresa al sistema, agrega los productos al carrito, visualiza el pedido. Después da clic en siguiente, donde ingresa nombre, sector, tipo de pago y da	productos seleccionados = Analgan, Diclofenaco cantidad = 1,1 pago=Efectivo Nombre=" Kerly" sector=Centro horario=hora actual	El usuario da clic en Confirmar Pedido y se muestra si está dentro del horario de atención.	El usuario da clic en Confirmar Pedido y se muestra si está dentro del horario de atención.

			<p>clic en <i>Confirmar pedido.</i></p> <p>Nota: El sistema compara si el pedido se realizó dentro del horario de atención.</p>			
5	Revisar la funcionalidad de obtener el recargo y total a pagar.					

Anexo 5: Taller de Caja blanca

Monopolio

Developers: Katty Desiderio Stephany Quimba

Taller de pruebas con travis y coveralls.

Coveralls es una herramienta que permite medir el porcentaje de cobertura del código de un programa por medio de tests. Se puede usar a través de GitHub, Bitbucket y Gitlab.

Para este taller realice lo siguiente:

Acepte la invitación del repositorio la cual fue enviada a su correo que vinculada su cuenta de github.

1. Realice un clone del repositorio:
`git clone https://github.com/stepquim/Monopolio.git`
2. Ingrese a la carpeta Monopolio:
`cd Monopolio`
3. Cree una nueva rama con el nombre de su usuario de github:
`git checkout -bsu_nombre_usuario_github`
4. En el archivo test.py del proyecto se encuentra codificada una prueba de caja blanca de sentencias. Codifique las pruebas necesarias para completar un porcentaje de cobertura de 100% (ramas), compruebe como el porcentaje de cobertura aumenta.
NOTA: SE RECOMIENDA USAR PYTHON 2.7.X RECUERDE COMENTAR CADA CASO DE PRUEBA QUE REALICE.
5. Corra localmente las pruebas y verifique que se ejecutan sin errores.
6. Realice el commit.
7. Haga el push sobre su rama: `git push origin su_nombre_usuario_github`(puede hacer tantos commits y push como sean necesarios hasta que todo esté correcto)

Nota: Para ver la ejecución de sus pruebas ir a <https://travis-ci.org/stepquim/Monopolio>

Nota: Para ver el porcentaje de cobertura de su rama ir a <https://coveralls.io/github/stepquim/Monopolio>.