

Universidad de las Ciencias Informáticas

Facultad 2



**Aplicación web para la promoción y
gestión de información de los artistas
de la agencia de representaciones
artísticas Clave Cubana**

Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas.

Autores:

Dianelys Díaz Arias

Eduardo Ángel Sedano Guzmán

Tutores:

Ing. Yenlys Guerra Dávila

Ing. Yaniel Alfredo Velázquez Bruceta

La Habana, Junio de 2016

Declaración de Autoría

Declaramos que somos los únicos autores del trabajo “**Aplicación web para la promoción y gestión de información de los artistas de la agencia de representaciones artísticas Clave Cubana**”. Autorizamos a la facultad 2 de la Universidad de las Ciencias Informáticas, y a la agencia de representaciones artísticas Clave Cubana, para que hagan el uso que estimen pertinente con el presente trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores

Dianelys Díaz Arias

Eduardo Ángel Sedano Guzmán

Tutores

Ing. Yenlys Guerra Dávila

Ing. Yaniel Alfredo Velázquez Bruceta

Datos de Contacto

Dianelys Díaz Arias

Correo: ddiaza@estudiantes.uci.cu

La Habana, Cuba

Eduardo Ángel Sedano Guzmán

Correo: easedano@estudiantes.uci.cu

La Habana, Cuba

Yenlys Guerra Dávila

Correo: yguerra@uci.cu

La Habana, Cuba

Yaniel Alfredo Velásquez Bruceta

Correo: ybruceta@uci.cu

La Habana, Cuba

Dedicatorias

Esta tesis va dedicada a mi madre. Por haberme dado tanto amor, y porque a pesar de que la vida no le alcanzó para verme en estos momentos decisivos, tengo la certeza de que aún me cuida desde el cielo y de que está conmigo a cada paso de mi camino. Todo mi amor para ella, mi ángel de la guarda.

Dianelys

A mamá por ser mi razón de ser cada día, por haber luchado en todo momento y por hacer de mi la persona que hoy soy.

A abuelo porque fuiste mi padre y mi ejemplo a seguir y donde quiera que estés hoy estarías orgulloso de mi.

A ti por compartir todos estos años de amor conmigo y sobrepasar toda adversidad.

Eduardo

Agradecimientos

Agradecemos a la universidad, por haber sido nuestro hogar por tanto tiempo, y a todo el personal docente que de una forma u otra hizo posible este momento.

A los tutores por su esfuerzo, dedicación y paciencia.

Al tribunal y a la oponente por sus consejos aportados para que el trabajo tuviera éxito.

Agradecemos al cliente por su colaboración y por haber confiado plenamente en nuestras capacidades como profesionales.

Autores

Agradezco en primer lugar a mi bella madre, gracias mami por hacer de mi la persona que soy, porque me distes amor, confianza, me enseñaste a estar preparada para los momentos difíciles de esta vida. Gracias porque a pesar de que hoy estás al ladito de Dios me sigues guiando y cuidando.

Agradezco a mi abuela Esther y a mi abuelo Oslay, que han sido mi fortaleza y mi sustento. A mi padre y su esposa Yanet por su apoyo y amor, a mis hermanos Luis, Lázaro, Jose y Lilian.

A mi guía, mi compañero, mi protector, mi gran amor Alejandro. Gracias mi vida por estos bellos años que hemos dejado de ser dos para convertirnos en uno. Gracias por no rendirte en los momentos difíciles, por no dejarme rendirme, gracias por haber sido mi fortaleza y mi inspiración. Gracias porque me haces mejor persona cada día.

Agradezco infinitamente a la familia de Alejandro, que ha dejado de ser su familia para ser nuestra familia, gracias por tanto amor y dedicación, en especial a Vilma, Fabiola y Tony.

Agradezco a mis grandes amigos, esos que, aunque pasen los años y cada quien haga su vida, siempre están conmigo. Ellos son Carla, Anne, Marilyn, Jhon Carlos y Oslaniel. Gracias por compartir el amor de sus familias conmigo y dejarme ser un miembro más de ellas. Gracias porque me hacen sentir dichosa en cuestiones de amistad.

Agradezco especialmente a Luisa, a quien quiero como a una madre, gracias a ella y a Carla, por su cariño incondicional.

Agradezco a Yanet y familia, quienes de una forma u otra están en mi vida desde que era una niña, gracias a Mirella por quererme como una hija, gracias a Yanet y a Liset por quererme como una hermana pequeña.

Gracias a mis compañeros y amigos de la universidad, gracias por hacerme tener las aventuras más hermosas para contar a mis nietos. En especial quiero agradecer a José Antonio, Carlos Rafael y Roniel Sánchez, porque a pesar de sacarme del paso muchas veces tengo la certeza de que me adoran.

Gracias a la profesora Yaniselis Sánchez, por su apoyo y amistad.

Agradezco especialmente a mi compañero de tesis, Eduardo, quien se ha convertido en un amigo. Gracias por su ayuda cuando la he necesitado, y por no estresarse conmigo cuando no nos hemos entendido.

Dianelys

Agradezco a mi mamá por haber estado dispuesta a dar la vida por mí aquel viernes a las 7:22am, por estar junto a mí cada segundo de mi vida, por luchar por mí y contra todas las adversidades, por darme fuerza de seguir adelante y lograr lo imposible. Por ser mi razón de ser cada día, mi ejemplo y guía a seguir. Agradezco por hacer de mí el hombre que soy y agradezco cada minuto el tenerte junto a mí. Estos momentos más que míos son tuyos.

A Any por estar junto a mí, todos estos años compartiendo los buenos y malos momentos. Por ser la mujer por la que late mi corazón en el amor y ayudarme a ver la vida de manera diferente. Por ser amiga y amante en todo momento y darme la fuerza de lograr algunas metas y planificar otras. Para ti hoy y entender que todos podemos lograrlo si así lo queremos, sin importar las dificultades.

Agradezco a mi abuelo Angel por haber sido mi padre y ejemplo de persona a seguir.

Donde quiera que se encuentre este día estaría orgulloso de mí.

Agradezco a mi hermano político por demostrarme el significado de la amistad, por confiar en mí por encima a veces de mí mismo y por enseñarme el arte de programar.

A mi hermano Lagdón por dotarme de los medios necesarios, dedicarme tiempo para algunos consejos sin importar la distancia y las diferencias.

A mi padre por haber aportado lo necesario (cadena genética) para que hoy este aquí y por haber dado rápidas enseñanzas en breves momentos y "conocer la persona que soy".

Agradecimientos

Agradezco a mi compañera de trabajo Dianelys por tener confianza en mí, sin casi conocerme, por apoyarme a pesar de mis tropiezos y depositar sobre mí la fe de que todo saliera bien cuando otros no lo hicieron. Especialmente muchas gracias por tu amistad.

Agradezco a mi discípulo y amigo por entender cada momento, por todas las aventuras que nos hemos permitido y por cada una de las diferencias que nos han aportado a ser mejores profesionales.

Agradezco a Yosvani por todo el tiempo y todas las dificultades pasadas que compartimos y nos han permitido ser mejores amigos.

Agradezco a Aymara por demostrarme el significado de nuestra amistad al pasar el tiempo, por las cosas vividas y por estar al pendiente de mí.

Agradezco el tiempo transcurrido con Carlos R, Roniel y Jose Antonio por las madrugadas y las aventuras de estos últimos tiempos.

Agradezco a todos mis compañeros de aula con los cuales he compartido todo este periodo y hemos creado lazos inquebrantables a pesar de las diferencias.

Eduardo

Resumen

Con el paso de los años, la promoción de productos de las industrias ha incrementado, debido a la necesidad de conquistar el mercado. Las industrias musicales no están exentas a esta actividad. En las agencias de representaciones artísticas la promoción constituye el elemento fundamental para la adquisición de clientes. La agencia de representaciones artísticas Clave Cubana tiene la misión de promocionar y comercializar un catálogo de proyectos artísticos de la música popular bailable. A pesar de la promoción que se realiza mediante los medios de difusión masiva, tales como periódicos, televisión o revistas, Clave Cubana cuenta con escasos clientes en el plano internacional. El fin de la presente investigación fue el desarrollo de una aplicación web para la promoción y gestión de información de los artistas de Clave Cubana. Para guiar el proceso del desarrollo se utilizó la metodología Programación Extrema, arrojando como resultado una aplicación web, que permite la promoción y gestión de información de los artistas representados por Clave Cubana.

Palabras Claves: aplicación web, información, módulo, promoción.

Índice

Introducción.....	XIII
Capítulo 1: Fundamentación teórica de la aplicación web propuesta.....	1
Introducción.....	1
1.1 Aplicaciones web.....	1
1.2 Promoción musical.....	1
1.3 Gestión de información.....	2
1.4 Antecedentes de las aplicaciones web para la promoción de artistas.....	2
1.5 Promoción de artistas musicales a través de la web. Tendencias actuales.....	4
1.6 Aplicaciones web para la promoción y gestión de información de los artistas.....	5
1.6.1 Internacionales.....	6
1.6.2 Nacionales.....	6
1.7 Metodología de desarrollo.....	7
1.8 Tecnologías y herramientas para el desarrollo.....	7
1.9 Conclusiones del capítulo.....	9
Capítulo 2: Características de la aplicación web propuesta.....	10
Introducción.....	10
2.1 Descripción general de la propuesta de solución.....	10
2.2 Fase exploración.....	13
2.2.1 Características no funcionales del sistema.....	14
2.2.2 Lista de funcionalidades del sistema.....	14
2.3 Fase planificación.....	15
2.3.1 Historias de Usuarios.....	15
2.3.2 Estimación de esfuerzo por historias de usuarios.....	18
2.3.3 Plan de iteraciones y entregas.....	19
2.4 Conclusiones del capítulo.....	21
Capítulo 3: Diseño e implementación de la aplicación web propuesta.....	22

Introducción	22
3.1 Arquitectura del Sistema	22
3.1.1 Flujo de procesos de la arquitectura del sistema.....	22
3.2 Patrón Arquitectónico.....	23
3.3 Patrones de diseño	24
3.3.1 Patrones GRASP	24
3.3.2 Patrones GoF	25
3.3.2.1 Creacionales	25
3.3.2.2 Estructurales	26
3.3.2.3 Comportamiento	26
3.4 Diseño de Clases de los módulos	26
3.5 Tarjetas CRC	28
3.6 Tareas de Ingeniería.....	31
3.7 Estándares de Codificación	35
3.8 Conclusiones del capítulo	38
Capítulo 4: Pruebas de la aplicación web propuesta	39
Introducción	39
4.1 Estrategia de Pruebas	39
4.1.1 Pruebas Unitarias	40
4.1.2 Pruebas de Aceptación	45
4.2 Aval de aceptación del cliente.....	56
4.3 Conclusiones del capítulo	57
Conclusiones Generales	58
Recomendaciones.....	59
Referencias bibliográficas	60
Bibliografía	62
Glosario de términos	64

Índice de Tablas

Tabla 1: Historia de usuario número 1: Gestionar galería de imágenes.....	17
Tabla 2: Historia de usuario número 2: Gestionar galería de videos.....	18
Tabla 3: Estimación de esfuerzos por historias de usuario.	19
Tabla 4: Plan de iteraciones.	20
Tabla 5: Plan de entregas.	21
Tabla 6: Tarjeta CRC de la clase RolController.	29
Tabla 7: Tarjeta CRC de la clase UserController.	30
Tabla 8: Tarjeta CRC de la clase RepositoryController.....	31
Tabla 9: TI 1. Listar galerías de imágenes.....	32
Tabla 10: TI 2. Adicionar galería de imágenes.	32
Tabla 11: TI 3. Modificar galería de imágenes.....	33
Tabla 12: TI 4.Eliminar galería de imágenes.	33
Tabla 13: TI 5. Mostrar galería de imágenes.	34
Tabla 14: TI 6. Listar artistas del catálogo.	34
Tabla 15: TI 7. Adicionar artista al catálogo.....	34
Tabla 16: TI 8. Modificar artista del catálogo.	35
Tabla 17: TI 9. Eliminar artista del catálogo.....	35
Tabla 18: TI 10. Mostrar artista del catálogo.....	35
Tabla 19: Caso de prueba de aceptación HU1_CP1.	46
Tabla 20:Caso de prueba de aceptación HU1_CP1.	47
Tabla 21:Caso de prueba de aceptación HU1_CP3.	48
Tabla 22:Caso de prueba de aceptación HU1_CP4.	49
Tabla 23:Caso de prueba de aceptación HU1_CP5.	50
Tabla 24:Caso de prueba de aceptación HU2_CP1.	51
Tabla 25:Caso de prueba de aceptación HU2_CP2.	52
Tabla 26:Caso de prueba de aceptación HU2_CP3.	53
Tabla 27:Caso de prueba de aceptación HU2_CP4.	54
Tabla 28:Caso de prueba de aceptación HU2_CP5.	55

Índice de Ilustraciones

Ilustración 1: Propuesta de solución.....	10
Ilustración 2: Interfaz gráfica del módulo portal.	12
Ilustración 3: Interfaz gráfica del blog del módulo portal.	12
Ilustración 4: Interfaz gráfica del módulo seguridad.....	13
Ilustración 5: Interfaz gráfica del módulo gestión.....	13
Ilustración 6: Arquitectura general del sistema Clave Cubana.....	22
Ilustración 7: Arquitectura MVC.....	23
Ilustración 8: Patrón creador.....	24
Ilustración 9: Patrón GOF Fábrica abstracta.....	25
Ilustración 10: Patrón GOF Singleton.....	26
Ilustración 11: Patrón GOF Decorador.....	26
Ilustración 12: Patrón GOF Iterativo.....	26
Ilustración 13: Fragmento del diseño de clases del módulo seguridad.....	27
Ilustración 14: Fragmento del diseño de clases del módulo gestión.....	27
Ilustración 15: Fragmento del diseño de clases del módulo portal.....	28
Ilustración 16: Estándar de codificación Identación.....	36
Ilustración 17: Estándar de codificación Comentarios.....	36
Ilustración 18: Estándar de codificación Declaración de variables.....	37
Ilustración 19: Estándar de codificación Declaración de métodos.....	37
Ilustración 20: Estándar de codificación Declaración de clases.....	37
Ilustración 21: Estándar de codificación de sentencias if y try/catch.....	38
Ilustración 22: Primera prueba unitaria aplicada a las funcionalidades de la iteración uno.....	40
Ilustración 23: Segunda prueba unitaria aplicada a las funcionalidades de la iteración uno.....	40
Ilustración 24: Primera prueba unitaria aplicada a las funcionalidades de la iteración dos.....	41
Ilustración 25: Segunda prueba unitaria aplicada a las funcionalidades de la iteración dos.....	41
Ilustración 26: Primera prueba unitaria aplicada a las funcionalidades de la iteración tres.....	42
Ilustración 27: Primera prueba unitaria aplicada a las funcionalidades de la iteración cuatro.....	42
Ilustración 28: Primera prueba unitaria aplicada a las funcionalidades de la iteración cinco.....	43
Ilustración 29: Primera prueba unitaria aplicada a las funcionalidades de la iteración seis.....	43
Ilustración 30: Primera prueba unitaria aplicada a las funcionalidades de la iteración seis.....	44
Ilustración 32: Primera prueba unitaria aplicada a las funcionalidades de la iteración siete.....	44
Ilustración 32: Segunda prueba unitaria aplicada a las funcionalidades de la iteración siete.....	44
Ilustración 33: No conformidades encontradas al realizar las pruebas unitarias por iteración.....	45
Ilustración 34: Aval de aceptación del cliente.....	56

Introducción

Desde los inicios de la humanidad, las actividades productivas estuvieron ligadas a la necesidad de comercializar los resultados de dicho proceso, exigiendo ineludiblemente que existiera algún medio que diera a conocer a las masas la existencia del producto y las vías de adquisición de este. Partiendo de esta idea es comprensible el hecho de que en las civilizaciones antiguas los medios para incentivar la compraventa de productos fueran los regalos, permitiendo a los artesanos y productores asegurar futuras inversiones por medio de obsequios finos y ornamentos especiales a los compradores que expendían grandes sumas de dinero (Santesmases, 1996).

En este punto es importante introducir el término promoción, que es fundamentalmente comunicación, traduciéndose como transmisión entre la entidad productora y la audiencia con el fin último de estimular la demanda (Santesmases, 1996). Con el paso de las épocas y la evolución de los medios y técnicas para promocionar artículos y servicios, la humanidad se va adentrando en los tiempos de auge de la comunicación, iniciando esta etapa con el surgimiento y desarrollo de los medios de difusión masiva tales como la radio y la televisión. Desde la primera etapa del siglo XX comienza a institucionalizarse la actividad promocional, estableciéndose parámetros y estándares para la realización de esta, como queda evidenciado en el Decreto Alemán sobre las Primas del 9 de marzo de 1932 (López, 2003).

Desde ese entonces la espiral evolutiva de la promoción ha ascendido vertiginosamente, condicionada fundamentalmente por los principios de la sociedad capitalista. El mayor auge comenzó a tener lugar a partir de la segunda Guerra Mundial, en los Estados Unidos. Este proceso se gestó debido a que la necesidad de conquistar el mercado fue mayor que la necesidad de producir en él, dado que las producciones se superaban, y quedaba excedente que no se podía vender, por el exceso de ofertas. A principios de la década del 50 del siglo pasado, la publicidad por sí sola no bastaba para vender un producto, consecuencia tanto del desarrollo de nuevas tecnologías como de la competitividad en el mercado. (López, 2003)

La publicidad en los medios de comunicación de masas resultaba insuficiente, por lo que se añade una sobresaturación publicitaria, la cual consistía en técnicas que ayuden a conocer, a motivar al consumidor y a empujarlo al acto de compra (López, 2003). Las actividades para la promoción no solo se aplicaron a los productos, sino también a los servicios ofertados por las empresas u organizaciones de otra índole dedicadas a distintas ramas productivas, entre ellas se situaban las pequeñas agrupaciones musicales. Estas últimas, promocionadas principalmente mediante pancartas y demostraciones.

Con el surgimiento de las Tecnologías de la Información y Comunicaciones (TIC), cambió la forma de crear, producir y generar música dentro de la industria musical. Pero también cambió la forma

de distribuirla (canales y soportes), comercializarla y consumirla, en gran medida producto al impacto de la mercadotecnia.

En la segunda mitad de la década del 90 del pasado siglo, comenzaron a darse en Cuba los primeros pasos para impulsar el uso de las TIC, con el propósito de aprovechar las ventajas y beneficios que ofrecen, en el desarrollo de la economía nacional y la sociedad. Fue así que comenzó en la Isla la carrera maratónica hacia la informatización (García, 2011). Proceso en el que quedan incluidas las empresas pertenecientes al Ministerio de Cultura, entre ellas la Sociedad Mercantil de Promociones Artísticas y Literarias (Artex S.A).

Artex S.A cuenta con un grupo de agencias, entre las que se encuentra la agencia de representaciones artísticas Clave Cubana que tiene la misión de promocionar y comercializar un catálogo de proyectos artísticos de la música popularailable (Artex.S.A). Gestiona presentaciones y contrataciones de agrupaciones musicales cubanas. A pesar de que en la actualidad la promoción se realiza mediante los medios de difusión masiva, tales como periódicos, televisión o revistas, Clave Cubana cuenta con escasos clientes en el plano internacional, limitando las contrataciones al plano nacional y por ende el desarrollo comercial de los artistas, provocando que estos busquen otras agencias para su representación.

Clave Cubana gestiona un gran cúmulo de información referente a los artistas: videos, imágenes, música y documentos. Todos estos archivos se encuentran guardados en formato digital y al ser tanta la información, se hace engorroso el proceso de búsqueda y actualización de estos archivos, sobre todo cuando un cliente necesite de ella. Actualmente cuando esto ocurre, sin importar la ubicación geográfica, los clientes deben dirigirse a la agencia, y luego de un extenso proceso de búsqueda de la información, pueden copiarla. Este proceso incomoda a los clientes, y algunos de ellos prefieren no hacer contratos con Clave Cubana, debido a que o no cuentan con los recursos necesarios para realizar el viaje, o no quieren agotarlos, por tanto, la agencia puede perder clientes.

Teniendo en cuenta lo expuesto anteriormente y la necesidad de solucionar estas dificultades, surge el siguiente **problema de investigación**: Actualmente en Clave Cubana, la promoción y gestión de información de los artistas está limitada, provocando un escaso número de clientes internacionales y haciendo engorroso los procesos de búsqueda y actualización de archivos.

Dicho problema se enmarca en el **objeto de investigación**: Los procesos de promoción de artistas y gestión de información.

Se define el **objetivo general** siguiente: Desarrollar una aplicación web para la promoción y gestión de información de los artistas de Clave Cubana, que apoye el aumento de clientes internacionales, y los procesos de búsqueda y actualización de archivos de la misma agencia.

Delimitando como **campo de acción**: Aplicaciones web para la promoción y gestión de información de artistas.

Con el objetivo de orientar el curso de la investigación, se descompone el problema a resolver en las siguientes **preguntas científicas**:

- ❖ ¿Cuáles son los principales fundamentos teóricos y metodológicos acerca de la promoción y gestión de información de artistas mediante el uso de aplicaciones web?
- ❖ ¿Qué componentes tendrá la aplicación web que contribuye a la promoción y gestión de información de artistas de Clave Cubana?
- ❖ ¿Qué resultados tendrá en la práctica la aplicación web desarrollada?

Para dar cumplimiento al objetivo general se deben realizar las siguientes **tareas de investigación**:

- ❖ Análisis crítico y valorativo de las aplicaciones web de promoción y gestión de información de los artistas, existentes a nivel nacional e internacional, estableciendo similitudes con la investigación en curso.
- ❖ Valoración de la metodología, tecnologías, librerías y herramientas en las que se realizará la solución del problema presentado.
- ❖ Análisis de los procesos de negocio asociados a la gestión de la información relacionada con la promoción de artistas, logrando un modelo único como guía para la implementación del sistema.
- ❖ Implementación de aplicación web aplicando las pautas de diseño definidas y siguiendo lo establecido en la lista de funcionalidades del software.
- ❖ Validación de la aplicación web aplicando la estrategia de prueba que propone la metodología de desarrollo que se seleccione.

Para desarrollar el presente trabajo se utilizaron los siguientes **métodos de investigación**:

Métodos Teóricos:

- ❖ Analítico-Sintético: Para descomponer el problema de investigación en elementos separados y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución propuesta.
- ❖ Inductivo-Deductivo: Para la identificación de la problemática y de las soluciones.
- ❖ Análisis-Histórico-Lógico: Para el estudio crítico de trabajos anteriores sobre la promoción y gestión de información de artistas, y para utilizar estos como punto de referencia y comparación con los resultados alcanzados.
- ❖ Modelación: Para modelar teóricamente la solución informática propuesta.

Métodos Empíricos:

- ❖ Observación: Para obtener la información necesaria para contestar la problemática y el problema de investigación acerca de las aplicaciones web para la promoción y gestión de información de artistas.

- ❖ Entrevista: Para obtener la información necesaria de los directivos de Clave Cubana antes de diseñar la solución informática propuesta y después de implantada la misma.

El presente trabajo de diploma está estructurado de la siguiente manera:

Capítulo 1: Fundamentación teórica de la aplicación web propuesta: Contiene el marco conceptual en el que se muestran las principales definiciones utilizadas en la investigación. Incluye un estudio de las aplicaciones web de promoción y gestión de información de artistas existentes en la actualidad, con el objetivo de seleccionar algunas de las funcionalidades que formarán parte de la propuesta de solución. En el capítulo se presenta la descripción de las tecnologías y herramientas a emplear para la solución del problema.

Capítulo 2: Características de la aplicación web propuesta: Describe la propuesta de solución. Se obtienen los artefactos de las dos primeras fases de la metodología XP, exploración y planificación, planteando las HU referente a cada funcionalidad del sistema, así como las tablas de estimaciones.

Capítulo 3: Diseño e implementación de la aplicación web propuesta: Se generan los artefactos de la III fase de la metodología XP, implementación, planteando las tarjetas CRC, las TI que describen las HU. Se presenta la arquitectura, se definen los patrones y estándares de codificación.

Capítulo 4: Pruebas de la aplicación web propuesta: Se ejecuta la cuarta fase de la metodología XP, pruebas, aplicando a la propuesta de solución pruebas unitarias y de aceptación. Luego de realizada esta fase se obtuvo un software probado, disminuyendo posibles errores y aumentando la calidad.

Capítulo 1: Fundamentación teórica de la aplicación web propuesta

Introducción

En el presente capítulo se lleva a cabo la fundamentación teórica de la aplicación web propuesta. Se tiene como principal objetivo, investigar, estudiar y analizar los principales elementos relacionados con las herramientas, metodologías y tecnologías a utilizar, para la creación de una posible solución al problema. Se tratan las principales definiciones de promoción y gestión de la información. Se realiza un estudio de las aplicaciones web que promocionan y gestionan información de artistas, para la recopilación de datos.

1.1 Aplicaciones web

Una aplicación web es un tipo especial de aplicación con la filosofía cliente/servidor, donde tanto el cliente (navegador) como el servidor (servidor web de aplicaciones) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de las aplicaciones. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero, además de la facilidad que ofrecen para llevar a cabo el proceso de mantenimiento sin necesidad de previa instalación. Se puede señalar además como ventaja, la posibilidad de acceso de usuarios concurrentemente al sistema. A continuación se mencionan las principales ventajas de las aplicaciones web; (Mora, 2002)

- ❖ Son actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesidad de afectar o interferir con sus hábitos de trabajo, debido a la no necesidad de nuevas descargas y procedimientos de instalación para acceder a últimas versiones.
- ❖ Tienen un precio menor que las aplicaciones de escritorio, ya que no requieren de la infraestructura de distribución, soporte técnico y marketing requerido por las aplicaciones descargables de software tradicionales.
- ❖ Los datos están online (en línea), es decir, no es necesario encontrarse precisamente en la computadora en la que se instaló el software para tener acceso al mismo.
- ❖ En el caso de las personas o instituciones que usan las aplicaciones web, encuentran en las mismas un medio de promoción rápido y eficiente.

1.2 Promoción musical

Según la RAE¹: **Promoción**, del latín *promotio*, es la acción y efecto de promover (impulsar un proceso o una cosa, tomar la iniciativa para realizar algo, elevar a alguien a un cargo superior al

¹ Real Academia Española.

que tenía). El término puede utilizarse para nombrar a las actividades que buscan dar a conocer o incrementar las ventas de algo.

Según Carles Martínez² la **promoción musical** es todos aquellos métodos que permitan a un músico difundir al máximo números de personas su música.

1.3 Gestión de información

La finalidad de la gestión de la información es ofrecer mecanismos que permitieran a la organización adquirir, producir y transmitir, al menor coste posible, datos e informaciones con una calidad, exactitud y actualidad suficientes para servir a los objetivos de la organización. En otros términos, sería conseguir la información adecuada, para la persona que lo necesita, en el momento que lo necesita, al mejor precio posible para tomar la mejor de las decisiones (Morales, 2004). Las funciones de la gestión de información se exponen a continuación: (Moreno, y otros, 2002)

1. Determinar las necesidades de información en correspondencia a sus funciones y actividades.
2. Mejorar los canales de comunicación y acceso a la información.
3. Mejorar los procesos informativos.
4. Empleo eficiente de recursos.

1.4 Antecedentes de las aplicaciones web para la promoción de artistas

Existe una estrecha relación entre las Industrias Culturales y las TIC, que más que una relación constituye una interdependencia pues, en la actualidad, es prácticamente imposible pensar en estas industrias sin utilizar las TIC o que las TIC se sigan desarrollando sin tener en cuenta las necesidades de las Industrias Culturales (Díaz, 2012).

Los ejemplos más triviales son las industrias del audiovisual (cine y televisión), la música y el libro, donde en todos los casos las TIC comenzaron interviniendo en el proceso productivo de estas industrias, pero terminaron formando parte de los procesos de distribución y consumo, así como de otros procesos de la cadena de valores. La Música fue la pionera de estos procesos y donde los esquemas industriales probaron los diferentes esquemas de producción y distribución. Los cambios afectaron las llamadas cadenas de valores de casi todos los negocios online, las formas del actual marketing digital, para el control y atención a clientes y los mecanismos de publicidad (Díaz, 2012).

El desarrollo de programas para generar y reproducir los archivos en formato MP3, hace que se tome como patrón y comienza a desarrollarse un mecanismo mediante el cual todo el que tuviera una computadora conectada a Internet podía compartir un espacio de su disco duro donde tuviera grabaciones musicales que quisiera compartir con otros interesados. Mediante un programa en la web se rastreaba lo que cada cual compartía, así lo ofrecía al resto de los interesados, incluyendo el control sobre las grabaciones repetidas y la posibilidad de copiar desde el lugar en el que en un

² Co-fundador de la web industrial musical y experto en el negocio de la música.

Fundamentación teórica de la aplicación web propuesta

momento determinado fuera más fácil y rápido para el interesado o continuar desde otro equipo si este primero fallaba. A esta clase de red se le conoce con el nombre de Peer to peer (Punto a punto) o por sus siglas P2P (Díaz, 2012).

En mayo de 1999, Shawn Fanning, de solo 19 años, junto a Sean Parker, fundan Napster, crean uno de estos servicios para compartir gratuitamente archivos de música a través de Internet. El proyecto es todo un éxito y alcanzó la cifra de 70 millones de usuarios, pero frente las demandas presentadas ante tribunales por diferentes demandantes del sector musical, en julio del 2001 Napster cierra sus puertas. El delito fue “infringir la ley de propiedad intelectual, por permitir el intercambio de archivos de música entre sus usuarios, sin control alguno, de manera gratuita” (Díaz, 2012).

La compañía fue adquirida por BMG que invirtió unos 85 millones de dólares en ella buscando convertirla en un sitio de venta de música en formato MP3, pero casi cae en bancarrota y es comprada por ROXIO, fabricante de programas para grabación de CD y DVD, a finales del 2002 por unos 5 millones de dólares (Díaz, 2012).

La industria de las TIC, ante la popularidad del formato MP3, venía trabajando en reproductores para este tipo de archivo musical y así marcaba el comienzo de este siglo por una gran avidez por los programas para convertir grabaciones musicales en archivos con formatos comprimidos como el MP3, circular los mismos a través de las redes y contar con aparatos específicos para su reproducción (Díaz, 2012).

El fenómeno “Napster”, replicado posteriormente en otros servicios como “Kazaa”, revolucionó por completo la industria del disco, abriendo el camino para una nueva forma de distribución y comercialización de las grabaciones musicales. Si bien propició el desarrollo de la piratería en la Red, demostró que existía una nueva forma de distribuir y consumir música, que debía ser tomada en cuenta seriamente (Díaz, 2012).

A pesar del fomento que esto generó en cuanto a la piratería, no eran los autores los que más protestaban por estos cambios, pues al final la mayoría de los autores nunca recibía lo que realmente le correspondía y vieron más en este proceso una posibilidad de promoción y romper anonimatos que de pérdida de beneficio (Díaz, 2012).

La verdadera protesta era liderada por los grandes consorcios de la industria musical que veían afectadas sus beneficios por esta aparente “revolución de los consumidores”. Muchos consideran que el real fomento de la piratería se produce al no asumir la industria estas nuevas formas de distribución y consumo. Es en este momento cuando se produce entonces un enfrentamiento entre el esquema clásico de distribución / comercialización / consumo frente al nuevo modelo que se desarrollaba con el apoyo de las TIC (Díaz, 2012).

En el 2003 uno de los líderes mundiales del desarrollo de las TIC, la empresa Apple creó un servicio en la web para comprar música en formato digital. La idea surge a partir de que esta compañía crea un reproductor de música digital denominado iPod. La iTunes Music Store (Almacenamiento de música iTunes) se convirtió en el “escaparate” más importante a nivel mundial para comercializar música digital! El panorama actual es bien distinto. Los reproductores de música digital en formato MP3 u otros son el medio por excelencia para consumir música, proceso que se ha extendido a otros equipos como las móviles (Díaz, 2012).

En la web son populares los sitios que comercializan o permiten el acceso a música en formato digital, tanto para escuchar, como para descargar hacia otro equipo. Incluso las formas de pagar por estos servicios han cambiado, existiendo diferentes variantes como pagar por descargar, pagar una cuota mensual por un grupo de servicios o sencillamente descargar o escuchar gratis (Díaz, 2012).

Hoy se estiman unos 13 millones de pistas licenciadas por compañías discográficas a los diferentes servicios de música existentes, que ya superan la cifra de 400. En el 2011 las ventas de música digital superaron el 30% de las ventas totales a nivel mundial, alcanzando la cifra de 4,6 mil millones y un crecimiento de los beneficios en más de un 6% con respecto al año anterior (Díaz, 2012).

Como elementos significativos podemos destacar que en los EE.UU. las ventas de digital superaron las ventas en físico (CD/DVD), y los principales espacios de publicidad comienzan a medir la popularidad de los artistas por la cantidad de descargas o reproducciones en servicios de reproducción (Díaz, 2012).

El uso de las TIC dentro de la industria musical cambió la forma de crear, producir, generar música, pero también ha cambiado la forma de distribuirla (canales y soportes), comercializarla y consumirla, en un dinámico proceso de cambio que aún no se detiene y que ha transformado los modelos económicos y comerciales que se aplican hoy al comercio online o comercio electrónico, no solo de la propia música y sus productos, sino del resto de los productos susceptibles de comercializar por esta vía (Díaz, 2012).

1.5 Promoción de artistas musicales a través de la web. Tendencias actuales

La coyuntura digital ha favorecido el surgimiento y desarrollo de multitud de portales y plataformas web para la distribución de música, desde iTunes a YouTube, pasando por muchas otras que ofrecen distintos tipos de acuerdos de distribución. El blog Aliadodigital.com en su informe “Distribución digital para artistas y discográficas independientes” señala que hay que estar muy atento a las condiciones y cláusulas que ofrecen algunos portales (por ejemplo, iTunes exige exclusividad, el mismo material no puede estar en otras distribuidoras), también señalan que hay

que estar siempre asesorado legalmente por expertos y que hay que intentar distribuir la música por el mayor número de tiendas on-line posibles. (Martínez, 2015)

Los secretos en cualquier campaña de promoción están en saber gestionar los contenidos y, sobre todo, en hacer una buena labor de relaciones públicas. Para conseguir que un medio de comunicación o una personalidad influyente tengan en consideración y hablen de una propuesta musical concreta, es indispensable disponer de contenidos que utilizar como mensaje. La música grabada, es el contenido más importante, pero otros contenidos como fotografías, vídeos, biografías, dossiers y notas de prensa resultan fundamentales. A ser posible, dichos contenidos han de ser originales, han de contar una historia y han de comunicar de forma clara quién es el artista o en qué consiste el evento musical que se está promocionando. (Martín, 2009)

Las relaciones públicas son las encargadas de engrasar el motor de cualquier campaña de promoción. Para promocionar a un artista se necesita saber quiénes son los líderes de opinión de su público objetivo, y saber dónde están y cómo localizarlos. Es importante también que las personas encargadas de difundir el mensaje sean capaces de convencer a otras personas de lo interesante que es dicho mensaje. Se puede hablar de marketing, redes sociales, discográficas, promotores, periodistas y perfiles de Facebook, pero al fin, se habla de comunicación y, por tanto, de personas. Por esta misma razón, la clave está en las relaciones públicas. (Martín, 2009)

Actualmente las grandes productoras, promotores o agencias de management (administración) suelen contar con sus propios medios de comunicación, a través de revistas online (en línea), radios digitales, portales web u otro tipo de medio en internet que permita la promoción de artistas. Pero para contar con una buena promoción no basta con que las agencias de management (administración) tengan un medio de comunicación, se debe garantizar que ese medio de comunicación sea atraído por los usuarios, y que la información que observen o lean sea de su interés, esto se logra realizando una adecuada gestión de información.

1.6 Aplicaciones web para la promoción y gestión de información de los artistas

Con el fin de identificar nuevas funcionalidades que podrían añadirse a las especificadas por el cliente para la propuesta de solución, se realiza un estudio de sistemas que promocionan y gestionan información de artistas. En la búsqueda realizada se encontraron muchas aplicaciones web, sin embargo, se seleccionaron para aportar a la investigación las que se muestran a continuación, debido a que son las que presentan características que se asemejan a las funcionalidades descritas por el cliente.

1.6.1 Internacionales

Deuvede Music: Es una asesoría artística especializada en promoción y marketing musical. El sistema que posee Deuvede cuenta con enlaces a redes sociales, en donde se publican noticias acerca de los artistas. Muestra el catálogo de los artistas que asesora.

Ángels MAS: Es una productora musical española. El sitio que contiene muestra el catálogo artístico de la productora, muestra a los usuarios noticias de los artistas, permite ver el contenido del sitio en dos idiomas. Presenta la información de contacto de la productora.

Tridimensional: Es una agencia que se dedica a la presentación de espectáculos de artistas musicales y bailarines. Cuenta con un sitio dedicado a la promoción de artistas que permite al usuario conocer los próximos eventos y noticias de la agencia. Muestra información de los artistas que promueve. Presenta la información de contacto.

Producciones Vikingas: Es una agencia que se dedica a la contratación de músicos. Cuenta con un sitio que posee el catálogo de los artistas, permite a los usuarios conocer las noticias de la agencia, tanto en el mismo sitio como en las redes sociales. Presenta información de contacto y los detalles para la contratación de artistas.

1.6.2 Nacionales

Musicalia: Agencia de representaciones artísticas de agrupaciones de variado formato de música popular bailable e instrumental, y produce espectáculos musicales, danzarios y circenses (Artex.S.A). El sistema que posee Musicalia permite realizar búsquedas y filtrado. Se pueden gestionar videos, imágenes, discografías. Presenta un catálogo de los artistas que promociona y muestra noticias.

BisMusic: Casa discográfica, distribuidora y editora musical que ofrece servicios fonográficos³ (Artex.S.A). El sistema que posee Bis Music cuenta con un catálogo de los artistas que promociona, está internacionalizado a dos idiomas y muestra noticias.

CubaMusic: Agencia que se dedica a la promoción de artistas cubanos. Muestra catálogo de los artistas que promociona, Permite a los usuarios ver las noticias mediante las redes sociales y en el propio sitio. Presenta canal RSS.

Tras el estudio de las aplicaciones informáticas expuestas, se arroja como resultado que, aunque todas llevan a cabo la promoción de artistas, no pueden ser reutilizadas debido a que no se tiene acceso al código fuente, por lo que queda descartada la posibilidad de reutilización. Solamente se enfocan en la promoción de artistas, dejando de lado la gestión de la información de los mismos. Sin embargo, el estudio de las aplicaciones web permitió identificar funcionalidades que se pueden utilizar en la propuesta de solución, tales como: Informar a los usuarios mediante noticias, enviar

³ Técnicas de reproducción y grabación de sonidos.

boletines de noticias haciendo empleo del canal RSS, mostrar videos e imágenes de los artistas, para consumo de los usuarios.

1.7 Metodología de desarrollo

La metodología para el desarrollo de software en un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado (Luis Calabria, 2003).

Según la filosofía de desarrollo se pueden clasificar las metodologías en dos grupos. Las metodologías tradicionales, que se basan en una fuerte planificación durante todo el desarrollo, y las metodologías ágiles, en las que el desarrollo de software es incremental, cooperativo, sencillo y adaptado (Luis Calabria, 2003).

Entre las metodologías de desarrollo ágiles más conocidas y usadas se encuentra la Programación Extrema (XP). Esta metodología plantea que se trabaje directamente con el cliente haciendo pequeñas iteraciones y como resultado mini entregas cada dos semanas, donde no existe más documentación que el propio código (Luis Calabria, 2003) .

Utiliza historias de usuarios (HU), las cuales tienen como objetivo dividir por partes la implementación de las funcionalidades de un software, donde se realicen en poco tiempo. Permiten el trabajo en parejas de programadores, donde uno aprende del otro y la obtención de resultados funcionales que se pueden ir evaluando y revisando (Luis Calabria, 2003).

La selección de XP como metodología de desarrollo de software se basa en los siguientes elementos: El equipo de desarrollo está formado por dos integrantes. El cliente está en constante comunicación con los desarrolladores, revisando cada funcionalidad al terminarse. Los especialistas de Clave Cubana no poseen conocimientos informáticos motivo por el cual las funcionalidades pueden variar.

1.8 Tecnologías y herramientas para el desarrollo

Para el desarrollo de la aplicación web se seleccionó un conjunto de tecnologías, compatibles con la política de software libre del país con el objetivo de lograr soberanía tecnológica. Para la selección de estas herramientas se debe tener en cuenta que la aplicación web a desarrollar debe ser desplegada en el servidor de Artex S.A, este utiliza **PHP**⁴ v5.4.12, como lenguaje de programación, servidor de aplicaciones **Apache** v2.4 y **MySQL** v.5.6 como sistema gestor de bases de datos, por

⁴ Procesador de Hipertexto

tanto, las herramientas que se seleccionen deben estar en función de las mencionadas anteriormente.

1.8.1 Herramienta de desarrollo

PHPStorm: Herramienta de desarrollo, se utiliza debido a que es un editor de texto para el lenguaje PHP, simplifica el trabajo para los programadores debido a que proporciona terminación inteligente de código al igual que otras herramientas. Soporta trabajos con la versión de PHP utilizada en la implementación. Por las razones mencionadas anteriormente se utiliza en la propuesta de solución y para generar los diagramas de clases del diseño. La versión usada es 2016.1.

1.8.2 Tecnologías del lado del servidor

WAMP⁵: El uso de este servidor web se debe a que como su nombre lo indica, es un sistema de infraestructura de internet que usa las herramientas siguientes: Windows como sistema operativo, Apache como servidor web, MySQL como gestor de base de datos y puede usar PHP como lenguaje de programación. Estas tres últimas tecnologías y herramientas son requeridas por el cliente, debido a que la aplicación a implementar debe adecuarse a las tecnologías usadas en el servidor en el que será desplegada. La versión utilizada es 2.4.

Symfony 2: Como plataforma de trabajo se utiliza symfony en su versión 2.7.6, por ser este un framework de PHP diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación. Además, automatiza las tareas comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (Potencier, y otros, 2009).

Doctrine: Para el acceso y las modificaciones de los datos se usa el ORM⁶: Doctrine⁷. De esta manera se puede acceder de forma efectiva a la base de datos desde un contexto orientado a objetos, ya que este framework traducirá la lógica de objetos a la lógica relacional y viceversa. Una de las ventajas principales que aporta el ORM es la reutilización, permitiendo citar a los métodos de un objeto de datos desde varias partes de la aplicación e incluso desde diferentes aplicaciones (Potencier, y otros, 2009).

Otra de las ventajas importantes de utilizar un ORM es la portabilidad, porque hace posible cambiar la aplicación a otra base de datos, incluso a mitad de desarrollo. Si se comenzara a desarrollar el proyecto con un gestor de base de datos determinado, por ejemplo, MySQL, Oracle o cualquier otro, bastaría cambiar una línea en un archivo de configuración y todo seguirá funcionando correctamente (Potencier, y otros, 2009). Desde la versión 1.3 de symfony, Doctrine se ha convertido oficialmente en su ORM por defecto.

⁵ Windows-Apache-MySQL-PHP/Python

⁶ Mapeo de objetos a base de datos relacional

⁷ Para mayor información visitar el sitio oficial de doctrine: [//www.doctrine-project.org/](http://www.doctrine-project.org/)

1.8.3 Tecnologías del lado del cliente

Para el desarrollo de las interfaces de usuario, se utiliza un conjunto de tecnologías que aumentan la interactividad, velocidad y usabilidad de las aplicaciones, entre ellas Ajax⁸ y JQuery como framework de JavaScript.

Ajax: No es una tecnología en sí mismo, sino que se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes. Las tecnologías que forman Ajax son las siguientes:

- ❖ XHTML⁹ y CSS¹⁰, para crear una presentación basada en estándares.
- ❖ DOM, para la interacción y manipulación dinámica de la presentación.
- ❖ XML¹¹, XSLT¹² y JSON¹³, para el intercambio y la manipulación asíncrona de información.
- ❖ JavaScript, para unir todas las demás tecnologías.

JQuery: Es una librería o framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML¹⁴, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología Ajax a páginas web. Al igual que otras librerías, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. Con las funciones propias de esta librería se logran resultados en menos tiempo y espacio (Project Team Of JQuery, 2010).

Bootstrap 3: Es un framework de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. Permitiendo a los desarrolladores la elaboración de un diseño que se adecue a las especificaciones del cliente.

1.9 Conclusiones del capítulo

El estado del arte permitió la identificación de nuevas funcionalidades que se pueden añadir a las entregadas por el cliente para implementar en la propuesta de solución. La investigación arrojó que la metodología más adecuada a utilizar dado las características del equipo de desarrollo y de la propuesta de solución a implementar es XP. Para la implementación de la propuesta de solución se seleccionaron un conjunto de tecnologías y herramientas en función de la infraestructura donde se va a desplegar.

⁸ Acrónimo de JavaScript asíncrono y XML

⁹ Lenguaje de marcado de hipertexto extensible

¹⁰ Hoja de estilo en cascada

¹¹ Lenguaje de marcas extensibles

¹² Lenguaje de transformación, hoja de estilo

¹³ Objeto de notación de JavaScript

¹⁴ Lenguaje de marcas de hipertexto

Capítulo 2: Características de la aplicación web propuesta

Introducción

En el presente capítulo se explica la propuesta de solución. Se desarrollan las fases exploración y planificación propuestas por la metodología XP. Se especifican las características no funcionales del sistema, y las funcionalidades mediante el empleo de HU. Se muestra la planificación de cada iteración en cuanto historias de usuario que las conforman y fechas de entrega.

2.1 Descripción general de la propuesta de solución

La propuesta de solución consiste en el desarrollo de una aplicación web que promocionará y gestionará la información de los artistas de Clave Cubana con el objetivo de apoyar el aumento de clientes internacionales y los procesos de búsqueda y actualización de archivos de la misma agencia. La aplicación web tendrá un control de acceso por roles. Esta aplicación web podrá ser accedida desde cualquier Cliente (navegador), permitiendo la interacción con los usuarios. La aplicación web contará con tres módulos como se puede observar en la ilustración 1.

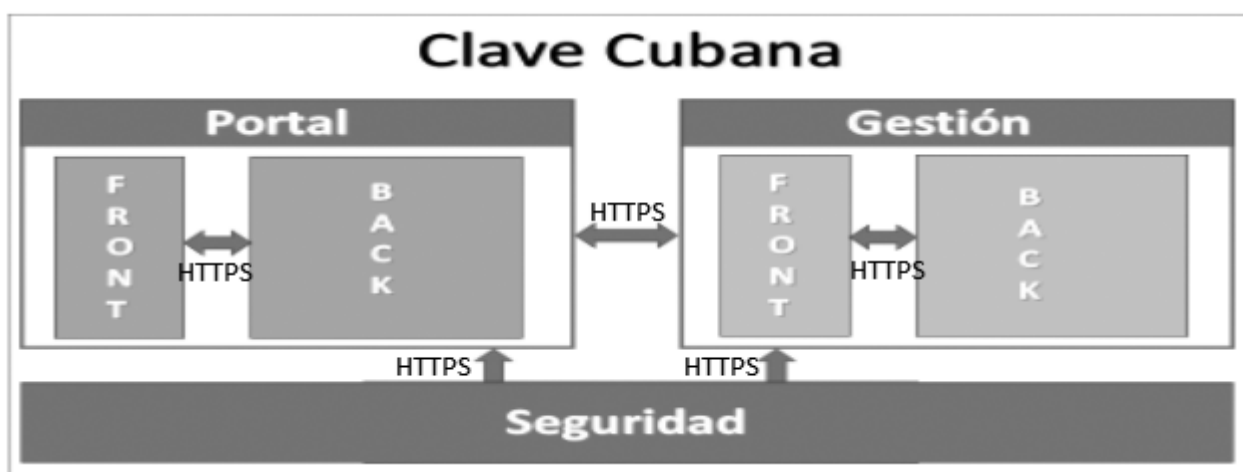


Ilustración 1: Propuesta de solución.

A continuación, se exponen los módulos que conformarán la aplicación web:

Portal: Es el módulo encargado de mostrar todo el contenido que se visualiza y con el que interactúa el usuario, proporcionando una visión empresarial de Clave Cubana en internet. El objetivo de la creación de este módulo es promocionar la información de los artistas, con tal fin se publican galerías de imágenes, álbumes musicales y noticias de los mismos. Contiene un blog que presenta artículos, los usuarios pueden realizar comentarios acerca de estos, logrando de esta forma la interacción dinámica. En el pie de página se muestra la información de contacto, enlaces a las redes sociales y al resto de las agencias que pertenecen a Artex. Contiene un espacio para que los usuarios seleccionen el idioma en que se visualiza el contenido.

Seguridad: Es el módulo encargado de suministrar la seguridad a toda la aplicación, para ello permite gestionar los usuarios con acceso al módulo portal y al módulo gestión, en dependencia del rol que se le asigne a cada uno. Permite además la visualización de los perfiles de usuario, el cambio de contraseña desde el portal para los roles usuario del sitio y cliente. Utiliza el controlador frontal que brinda el framework para el trabajo con sesiones.

Gestión: Es el módulo que le permite al administrador gestionar la información a publicar en el portal. Le brinda las opciones de eliminar comentarios realizados por el resto de los usuarios, además de poder aceptar o no los usuarios del sistema y asignarle un rol a cada uno de ellos.

A continuación, se exponen los roles definidos en la aplicación web propuesta como solución:

- ❖ **Rol anónimo:** Los usuarios con este rol solamente pueden visualizar la información que se muestra en la portada del módulo portal.
- ❖ **Rol usuario del sitio:** Este rol es un rol anónimo a la vez, pero el usuario tiene permiso para comentar en el blog, realizar acciones de like (gustar) y dislike (no gustar) a la información del portal.
- ❖ **Rol cliente:** Este rol es a la vez un rol usuario de sitio. Es el rol definido para visualizar la información técnica y profesional de interés para la empresa, tal como las necesidades técnicas de los escenarios para las presentaciones de los artistas.
- ❖ **Rol administrador:** El usuario con este rol asignado es el encargado de administrar todo el contenido de la aplicación web.

Características de la aplicación web propuesta

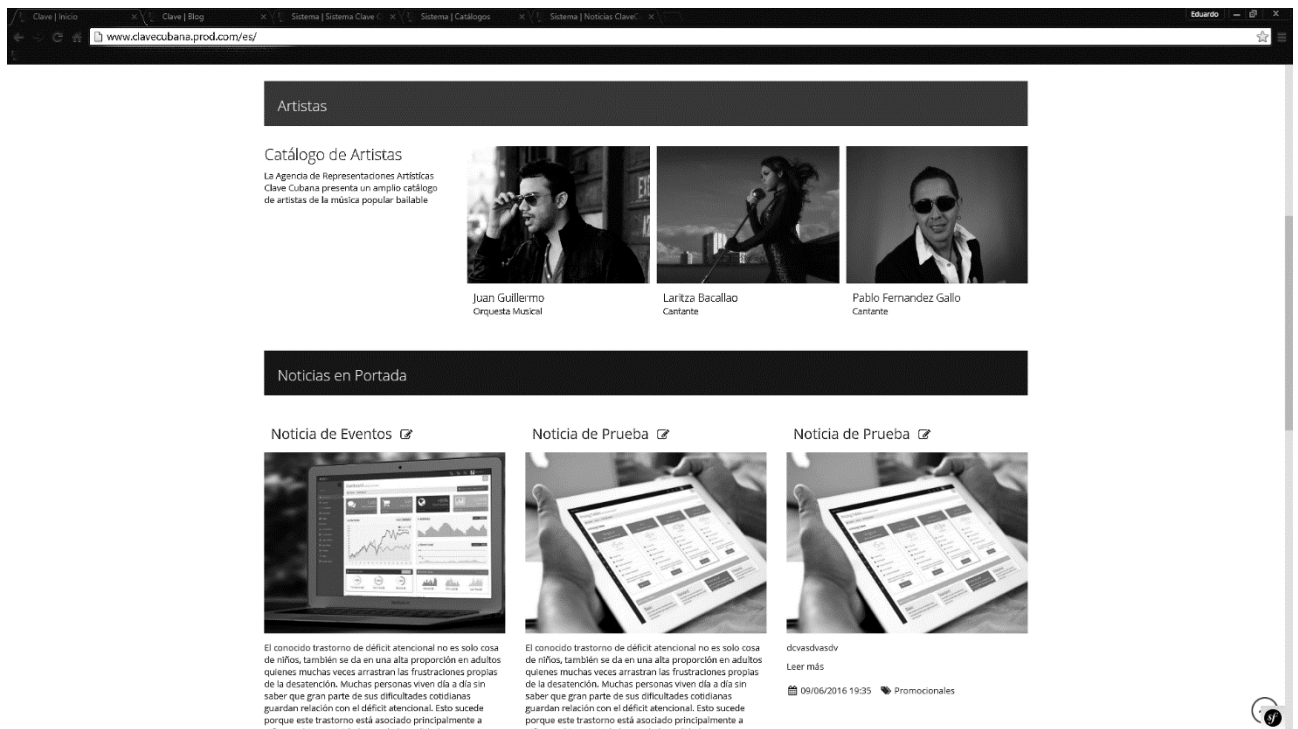


Ilustración 2: Interfaz gráfica del módulo portal.

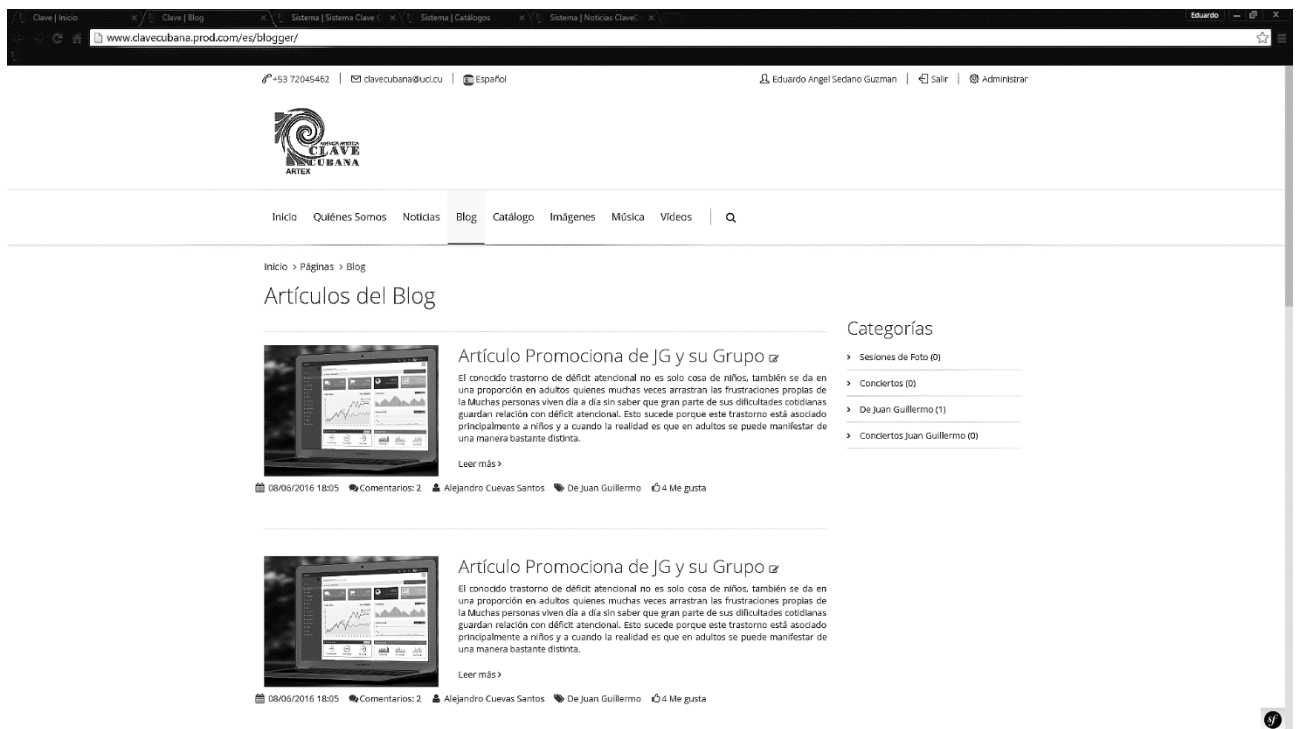


Ilustración 3: Interfaz gráfica del blog del módulo portal.

Características de la aplicación web propuesta

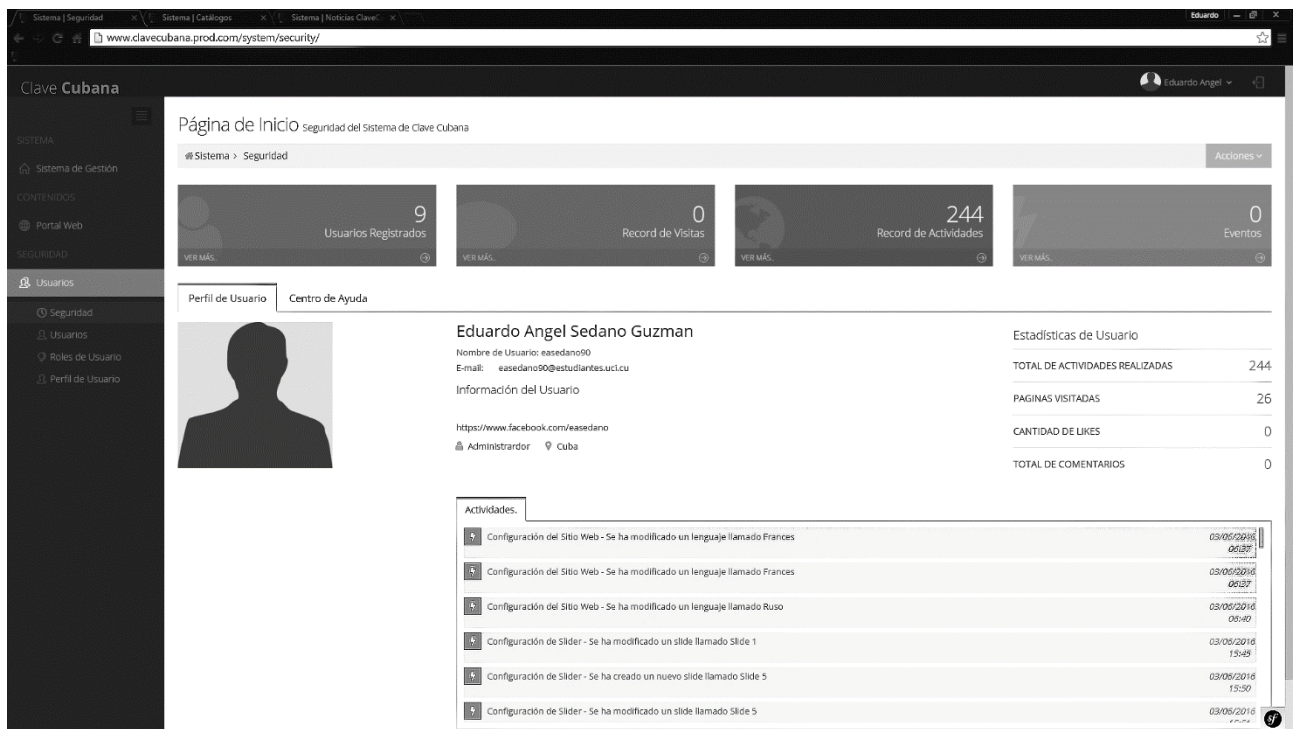


Ilustración 4: Interfaz gráfica del módulo seguridad.

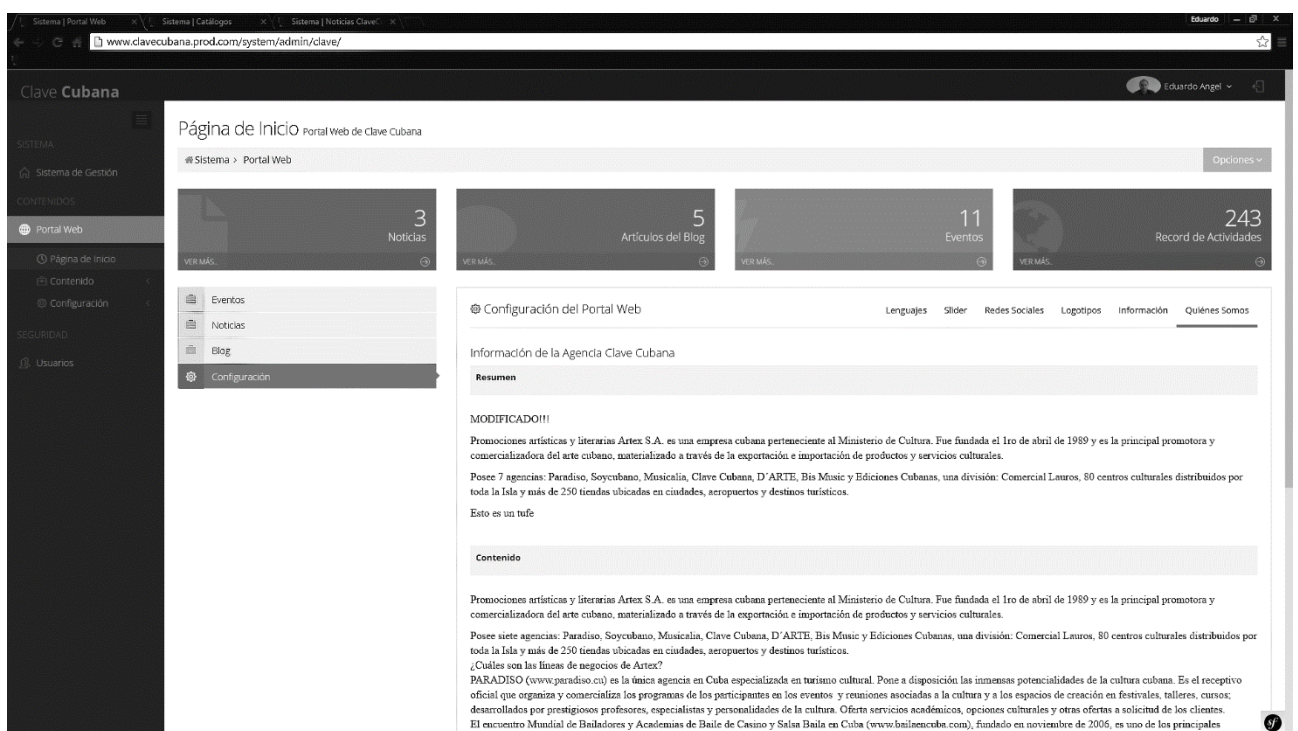


Ilustración 5: Interfaz gráfica del módulo de gestión.

2.2 Fase exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

2.2.1 Características no funcionales del sistema

Las características no funcionales son rasgos que se desean del sistema, señalan una restricción del mismo y son fundamentales en el éxito del producto, son propiedades o cualidades que el producto debe tener y normalmente se vinculan a funcionalidades (Luis Calabria, 2003).

- ❖ Apariencia o interfaz externa: Debe seguir las pautas del diseño del sitio corporativo de Artex. Debe presentar una interfaz en tonos claros y resaltar los colores identificativos de Clave Cubana (rojo y azul). Mostrar el logotipo de la agencia así como los logotipos del resto de las agencias pertenecientes a Artex, según el manual de identidad.
- ❖ Usabilidad: La aplicación web Clave Cubana podrá ser usada por cualquier tipo de usuario. Los mensajes de error deben ser lo suficientemente informativos y precisos. Los elementos de navegación deberán ser orientados al usuario.
- ❖ Seguridad: Debe cumplir con los pilares de la seguridad informática siguientes:
 1. Confidencialidad: La información de la aplicación web Clave Cubana debe ser accedida solamente por las personas que tienen autorización para hacerlo. Los usuarios que interactúen con ella pueden acceder a la información según los roles asignados.
 2. Disponibilidad: Debe tener métodos de precaución contra posibles daños, tanto en la información como en el acceso a la misma, para evitar posibles bloqueos.
 3. Autenticidad: La aplicación web Clave Cubana debe tener método de autenticidad para garantizar que los datos sean reales.
- ❖ Legales: La aplicación web Clave Cubana debe ser desarrollado con tecnologías libres, lo que permite independencia tecnológica.
- ❖ Software: Navegador web (Mozilla Firefox v30 o superior, Google Chrome v50 o superior) para el acceso a la plataforma y conexión de red.
- ❖ Hardware: Computadora de dos GB de RAM o superior. Espacio en disco de 100GB como mínimo.

2.2.2 Lista de funcionalidades del sistema

En XP se elimina la fase inicial de captura de requisitos y se permite que éstos se vayan definiendo de una forma ordenada durante el tiempo que dura el proyecto. El cliente puede cambiar de opinión sobre la marcha y a cambio debe encontrarse siempre disponible para resolver dudas del equipo de desarrollo y para detallar los requisitos especificados cuando sea necesario. El proceso de captura de requisitos de XP gira entorno a una lista de características que el cliente desea que existan en el sistema final (Fuentes, 2012). A continuación, se listan las funcionalidades del sistema, aunque se puedan observar en un sentido muy amplio, quedarán definidos de forma específica en las tareas de ingeniería.

1. Gestionar galería de imágenes.

2. Gestionar catálogo de artistas.
3. Gestionar álbum musical.
4. Gestionar galería de videos.
5. Gestionar cargador de archivos.
6. Gestionar redes sociales.
7. Gestionar lenguajes.
8. Gestionar perfil de usuario.
9. Gestionar usuarios con acceso al sistema.
10. Gestionar categoría de artistas.
11. Gestionar roles de usuarios.
12. Gestionar noticias.
13. Gestionar categorías de noticias.
14. Gestionar artículos del blog.
15. Gestionar categorías de los artículos del blog.
16. Gestionar comentarios de los artículos del blog.
17. Gestionar logotipos.
18. Gestionar información de la empresa e información de contacto.

2.3 Fase planificación

La segunda fase de la metodología XP es Planificación, la cual comienza con una serie de HU, que se encargan de describir las características y funcionalidades del sistema que se quiere construir. Durante esta fase se estimará el esfuerzo de implementar cada HU. También se calcula el tiempo de cada iteración para el desarrollo del sistema.

2.3.1 Historias de Usuarios

La metodología XP utiliza la técnica de las HU para sustituir a los documentos de especificación funcional y a los casos de uso. Estas HU son escritas por el cliente en su propio lenguaje como descripciones cortas de lo que el sistema debe realizar. El tratamiento de las HU es muy dinámico y flexible, permite que en cualquier momento se puedan romper, reemplazar por otras más específicas o generales, añadirse nuevas o ser modificadas. Para ser implementadas las HU, el cliente y los desarrolladores se reúnen para detallar las funcionalidades de cada una. El tiempo de desarrollo ideal para una HU varía entre una y tres semanas (Penadés, 2010).

Según Kent Beck cada HU recoge los siguientes aspectos:

Número: Número asignado a la HU.

Nombre de HU: Atributo que contiene el nombre de la HU.

Usuario: El usuario del sistema que utiliza o protagoniza la HU.

Características de la aplicación web propuesta

Prioridad en el negocio: Contiene el nivel de prioridad de la HU en el negocio. Es alta en caso de que la HU sea indispensable en el negocio, media en caso de que su realización o no afecte el negocio y baja cuando no se considera una prioridad.

Riesgo de desarrollo: Contiene el nivel de riesgo en caso de no realizarse la HU. Es alta, si el riesgo de no realizar la HU incide en el funcionamiento de la plataforma, media si el riesgo de no realizarla es medianamente importante, y baja en caso de que no se considere un riesgo tardar en la realización de la HU y no incida en el funcionamiento de la plataforma.

Puntos estimados: Este atributo es una estimación hecha por el equipo de desarrollo sobre el tiempo de duración de la HU. Cuando el valor es uno equivale a una semana ideal de trabajo, y un día equivale a 0,2 puntos.

Iteración asignada: Especifica la iteración a la que pertenece la HU correspondiente.

Descripción: Posee una breve descripción de lo que realizará la HU.

Luego de analizar los datos y especificaciones de cada campo comprendido en el modelo de HU se obtuvo para el presente trabajo un total de 18 HU. Cada una se corresponde con las funcionalidades expuestas anteriormente. A continuación, se muestran las historias de usuarios de la iteración número 1 del desarrollo del software. El resto de las historias de usuarios se exponen en el [Anexo I](#) del presente trabajo.


Historia de Usuario	
No: 1	Usuario: Administrador del sistema Clave Cubana.
Nombre de la HU: Gestionar galería de imágenes.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Media
Puntos Estimados: 1,4	Iteración Asignada: 1
Programador Responsable: Dianelys Díaz Arias y Eduardo Ángel Sedano Guzmán	
Observaciones: El administrador del sistema debe estar autenticado.	
Descripción: El usuario puede listar, adicionar, eliminar, mostrar y modificar la galería de imágenes.	
Prototipo de Interfaz:	
	

Tabla 1: Historia de usuario número 1: Gestionar galería de imágenes.

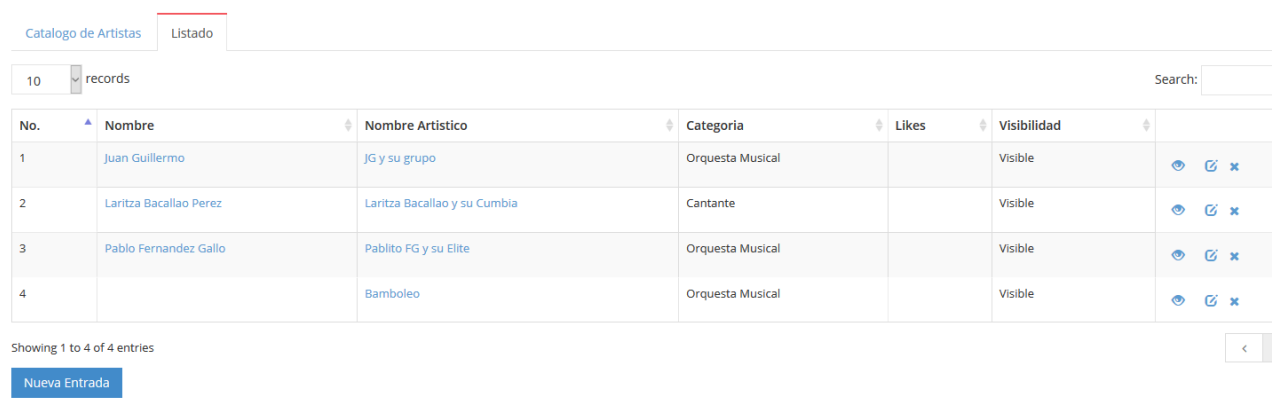
Historia de Usuario																																				
No: 2	Usuario: Administrador del sistema Clave Cubana.																																			
Nombre de la HU: Gestionar catálogo de artistas.																																				
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Media																																			
Puntos Estimados: 1,4	Iteración Asignada: 1																																			
Programador Responsable: Dianelys Díaz Arias y Eduardo Ángel Sedano Guzmán																																				
Observaciones: El administrador debe estar autenticado.																																				
Descripción: El usuario puede listar, adicionar, eliminar, mostrar y modificar los catálogos de los artistas.																																				
Prototipo de Interfaz:																																				
 <p>The screenshot shows a web interface for managing artists. At the top, there are two tabs: 'Catalogo de Artistas' (active) and 'Listado'. Below the tabs is a search bar and a dropdown menu set to '10 records'. The main content is a table with the following columns: 'No.', 'Nombre', 'Nombre Artístico', 'Categoría', 'Likes', 'Visibilidad', and a set of action icons (eye, edit, delete). The table contains four rows of data:</p> <table border="1"> <thead> <tr> <th>No.</th> <th>Nombre</th> <th>Nombre Artístico</th> <th>Categoría</th> <th>Likes</th> <th>Visibilidad</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Juan Guillermo</td> <td>JG y su grupo</td> <td>Orquesta Musical</td> <td></td> <td>Visible</td> <td>👁️ ✎️ ✖️</td> </tr> <tr> <td>2</td> <td>Laritza Bacallao Perez</td> <td>Laritza Bacallao y su Cumbia</td> <td>Cantante</td> <td></td> <td>Visible</td> <td>👁️ ✎️ ✖️</td> </tr> <tr> <td>3</td> <td>Pablo Fernandez Gallo</td> <td>Pablito FG y su Elite</td> <td>Orquesta Musical</td> <td></td> <td>Visible</td> <td>👁️ ✎️ ✖️</td> </tr> <tr> <td>4</td> <td></td> <td>Bamboleo</td> <td>Orquesta Musical</td> <td></td> <td>Visible</td> <td>👁️ ✎️ ✖️</td> </tr> </tbody> </table> <p>Below the table, it says 'Showing 1 to 4 of 4 entries' and there is a 'Nueva Entrada' button.</p>		No.	Nombre	Nombre Artístico	Categoría	Likes	Visibilidad	Acciones	1	Juan Guillermo	JG y su grupo	Orquesta Musical		Visible	👁️ ✎️ ✖️	2	Laritza Bacallao Perez	Laritza Bacallao y su Cumbia	Cantante		Visible	👁️ ✎️ ✖️	3	Pablo Fernandez Gallo	Pablito FG y su Elite	Orquesta Musical		Visible	👁️ ✎️ ✖️	4		Bamboleo	Orquesta Musical		Visible	👁️ ✎️ ✖️
No.	Nombre	Nombre Artístico	Categoría	Likes	Visibilidad	Acciones																														
1	Juan Guillermo	JG y su grupo	Orquesta Musical		Visible	👁️ ✎️ ✖️																														
2	Laritza Bacallao Perez	Laritza Bacallao y su Cumbia	Cantante		Visible	👁️ ✎️ ✖️																														
3	Pablo Fernandez Gallo	Pablito FG y su Elite	Orquesta Musical		Visible	👁️ ✎️ ✖️																														
4		Bamboleo	Orquesta Musical		Visible	👁️ ✎️ ✖️																														

Tabla 2: Historia de usuario número 2: Gestionar catálogo de artistas.

2.3.2 Estimación de esfuerzo por historias de usuarios

La metodología XP en sus principios generales establece que cada una de las historias de usuario necesitan de una a tres semanas de desarrollo. Las estimaciones de esfuerzo se realizan en una escala de uno a tres puntos, siendo un punto la referencia a una semana ideal de programación. En la siguiente tabla se muestran los puntos estimados de cada historia de usuario.

No	Historia de usuario	Puntos de estimación
1	Gestionar galería de imágenes.	1,4
2	Gestionar catálogo de artistas	1,4
3	Gestionar álbum musical.	1,4
4	Gestionar galería de videos.	1,4
5	Gestionar cargador de archivos.	1,2
6	Gestionar redes sociales.	1,2
7	Gestionar lenguajes.	1,2
8	Gestionar perfil de usuario.	1,2
9	Gestionar usuarios con acceso al sistema.	1,2
10	Gestionar categorías de artistas.	0,8
11	Gestionar roles de usuarios.	0,8
12	Gestionar noticias.	0,8
13	Gestionar categorías de noticias.	0,8
14	Gestionar artículos del blog.	0,8
15	Gestionar categorías de los artículos del blog.	0,6
16	Gestionar comentarios de los artículos del blog.	0,6
17	Gestionar logotipos.	0,6
18	Gestionar información de la empresa e información de contacto.	0,6

Tabla 3: Estimación de esfuerzos por historias de usuario.

2.3.3 Plan de iteraciones y entregas

Para seleccionar las historias de usuarios a desarrollar en cada iteración, se sigue el orden de prioridad del negocio de cada una de ellas, desarrollando primero las de prioridad alta, seguidas por las de prioridad media y por último las de prioridad baja. Se definieron un total de siete iteraciones, siguiendo los criterios de duración de iteración que propone XP, el cual plantea que cada iteración tiene una duración de una a tres semanas. En cada iteración se realizan los test funcionales para cada una de las historias a ser implementadas. Al final de cada iteración el cliente debe analizar que todas las historias estén implementadas. Debe también correr todos los test funcionales y que estos resulten ser exitosos (Penadés, 2010).

Siempre que se realice una iteración se define una nueva versión del sistema. En la tabla 6 se muestra el plan de iteraciones, donde se representa cada una de las 7 iteraciones y las HU asociadas a cada una de estas.

Características de la aplicación web propuesta

Iteración	Historias de Usuarios a Implementar	Duración en semanas/ días de la iteración
1	Gestionar galería de imágenes. Gestionar catálogo de artistas.	2,8
2	Gestionar álbum musical. Gestionar galerías de videos.	2,8
3	Gestionar cargador de archivos. Gestionar redes sociales.	2,4
4	Gestionar lenguajes. Gestionar perfil de usuario.	2,4
5	Gestionar usuarios con acceso al sistema. Gestionar categorías de artistas. Gestionar roles de usuarios.	2,8
6	Gestionar noticias Gestionar categorías de noticias. Gestionar artículos del blog. Gestionar categorías de los artículos del blog.	3
7	Gestionar comentarios de los artículos del blog. Gestionar logotipos. Gestionar información de la empresa e información de contacto.	1,8

Tabla 4: Plan de iteraciones.

En el plan de entregas se establecen las fechas en las que se publicarán las versiones del sistema. La primera iteración comienza el 28 de enero del 2016 y la fecha planificada de entrega para la última versión es el 2 de junio del 2016. En la tabla 7 se pueden observar las fechas de entrega de cada versión de la aplicación web para la promoción y gestión de la información de los artistas de Clave Cubana.

Características de la aplicación web propuesta

Fecha de entrega	Iteración
16 de febrero del 2016	1
7 de marzo del 2016	2
24 de marzo del 2016	3
11 de abril del 2016	4
29 de abril del 2016	5
20 de mayo del 2016	6
2 de junio del 2016	7

Tabla 5: Plan de entregas.

2.4 Conclusiones del capítulo

En el presente capítulo se determinó como propuesta de solución un sistema compuesto por tres módulos, portal, gestión y seguridad. Se obtuvieron los artefactos correspondientes a las fases de exploración y planificación.

Capítulo 3: Diseño e implementación de la aplicación web propuesta

Introducción

En el presente capítulo se describe la fase III de la metodología XP, correspondiente al diseño e implementación. Se presenta la arquitectura, se definen los patrones y estilos de codificación utilizados, además se representan las tarjetas CRC. También se especifican las tareas de ingeniería para describir en detalles cada una de las HU.

3.1 Arquitectura del Sistema

Mediante la arquitectura general de Clave Cubana, se puede mostrar de forma más organizada el flujo de actividades que se realizan en el proceso de petición-respuesta, dando a comprender a mayor escala la influencia de la propuesta de solución en dichas actividades.

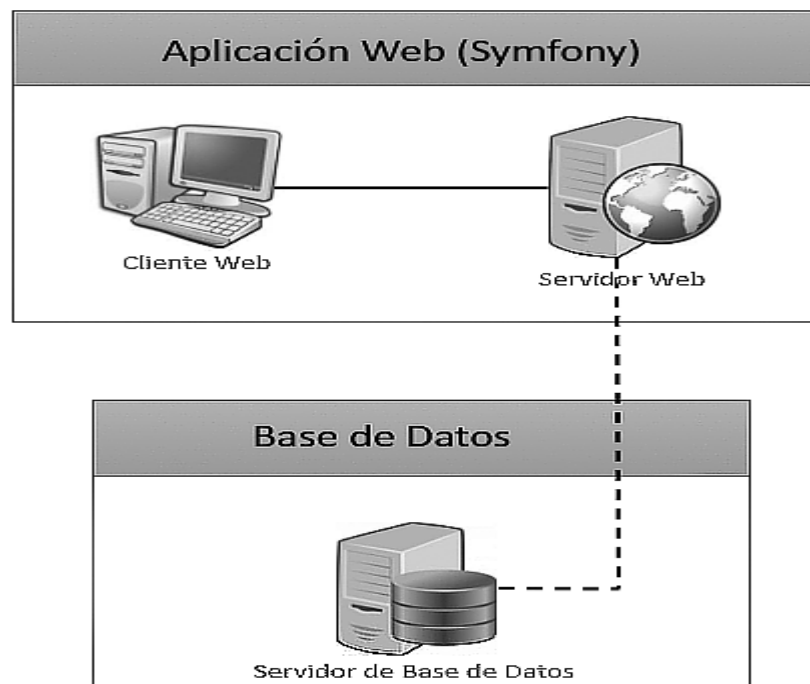


Ilustración 6: Arquitectura general del sistema Clave Cubana.

Detalles de los componentes utilizados en la representación de la arquitectura:

————— Relación de un componente con otro.

----- Proceso interno que guarda relación con un componente.

3.1.1 Flujo de procesos de la arquitectura del sistema

1. Cliente web-Servidor web: El cliente web realiza una petición al servidor web. El controlador frontal del framework symfony *HttpKernel* recibe dicha petición y la procesa.

2. Servidor web- Servidor de Base de Datos: El controlador frontal del framework symfony obtiene una abstracción orientada a objeto de la petición enviada por el cliente web. Dicha clase delega la selección del controlador a ejecutar a una implementación de *HttpKernel\Controller\ControllerResolverInterface*. Cuando es ejecutada dentro del controlador la acción a la que hace referencia la petición procede mediante el ORM Doctrine a interactuar con los datos.

3. Servidor de Base de Datos-Servidor web: Cuando es ejecutada por el ORM Doctrine una consulta a la base de datos se obtiene una abstracción orientada a objetos de los datos, los mismos son procesados en la acción ejecutada en el controlador seleccionado.

4. Servidor web-Cliente web: Al ser procesados los datos en la acción ejecutada en el controlador seleccionado, son transformados por el controlador frontal y enviados para mostrarlos en el cliente web mediante el motor de plantilla Twig.

3.2 Patrón Arquitectónico

Consiste en una estructura que plantea la forma en que se van a organizar los componentes de un sistema de software y las responsabilidades de cada uno. Se seleccionó para utilizar el patrón arquitectónico Modelo Vista Controlador (MVC), porque entre las tecnologías que sustentan la propuesta de solución se encuentra el framework Symfony2, el cual emplea dicho patrón, por consiguiente, se pretende extender su utilización en el proceso de desarrollo.

Esta arquitectura permite dividir la aplicación en tres grandes capas: el modelo (Model), define la lógica de negocio (la base de datos pertenece a esta capa), la vista (View), es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa) y el controlador (Controller), que es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario (Fuentes, 2012).

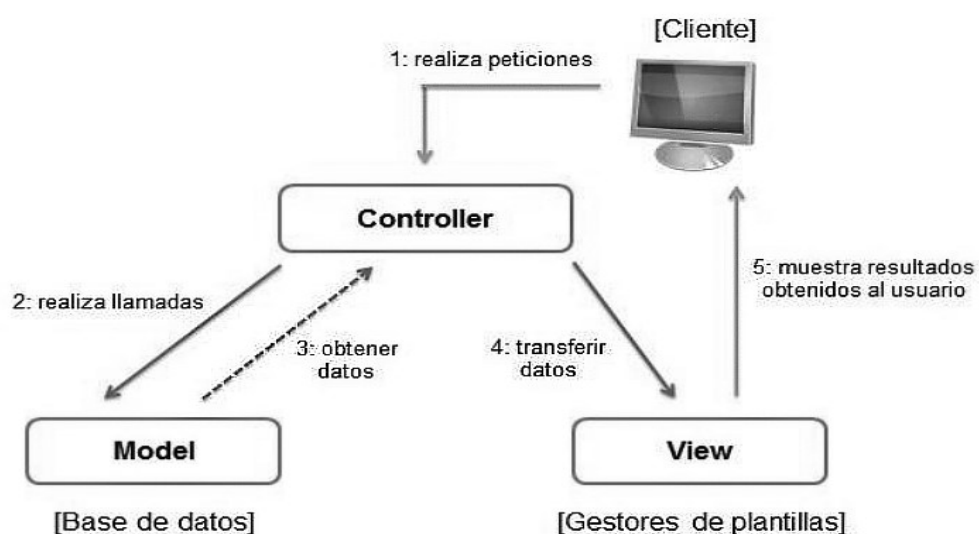


Ilustración 7: Arquitectura MVC.

Como se muestra en la ilustración 7 el cliente (navegador) realiza las peticiones a través de la interfaz gráfica, las cuales son interceptadas por el controlador quien realiza las validaciones necesarias, procesamiento de dichos datos y lógica de negocio asociadas a esa petición del cliente. El controlador envía los datos al modelo, para ser guardados en una base de datos y/o los obtiene dependiendo de la solicitud del usuario para finalmente enviarlos a la vista a fin de ser mostrados nuevamente al cliente a través de una respuesta.

3.3 Patrones de diseño

Según Nicolás Tedeschi ¹⁵ los patrones de diseño son considerados soluciones ya probadas a problemas de desarrollo de software sujeto a contextos similares. No se utilizan arbitrariamente, se debe tener en cuenta el nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). Para la presente investigación se hizo uso de los patrones que a continuación se mencionan:

3.3.1 Patrones GRASP

Los patrones GRASP se consideran una serie de “buenas prácticas” para el diseño de software. El objetivo fundamental de su utilización fue definir la clase con mayor jerarquía e implementar en ella los métodos necesarios, además de asignar responsabilidades a las clases dependientes.

Experto: Este patrón se evidencia en las clases modelos del sistema, debido a que estas son las encargadas de realizar la abstracción de datos y consultas a base de datos, además de presentar todos los atributos necesarios para modelar el comportamiento de las entidades.

Creador: Se utilizó este patrón en las clases controladoras para trabajar con objetos instanciados en las clases que representan las entidades. Por ejemplo, en la clase controladora *UserController* se implementa el método *createCreateForm* para la creación de nuevos formularios, siendo el creador *createForm*, tal y como se muestra en la ilustración 8.



Ilustración 8: Patrón creador.

Bajo acoplamiento: Las clases que realizan las acciones ejecutadas por el sistema en su definición heredan de la clase *Controller*. Permittedo lograr una jerarquía de clases, donde todas presentan

¹⁵ Nicolás Tedeschi: Analista de sistemas de la facultad de ingeniería de Uruguay. Desde 1995 trabaja con tecnologías Microsoft de tres capas (HTML-ASP, COM y SQL Server). Es además desarrollador 5 estrellas de Microsoft MSDN y ha obtenido 3 estrellas de ese programa.

un mismo padre, pero con implementaciones diferentes, de forma tal que si se modifican los datos de alguna de las clases, no afecta al resto.

Alta Cohesión: El uso de este patrón permitió que las tareas trabajen sobre los mismos objetos pero que tengan su codificación independiente y sus propias responsabilidades. Por ejemplo las clases controladoras *ImageGalleryController* y *ImageController* utilizan los objetos *ImageGallery* e *Image*, sin embargo cada una tiene sus propias funciones e implementaciones, lo que permite que la complejidad sea lo más manejable posible y que se adapte mejor en casos de reutilización.

Controlador: Al utilizar el framework symfony, se extiende el uso del patrón controlador, debido a que la arquitectura que propone el framework se basa en el modelo, la vista y el controlador. Por tanto, todas las clases que separan la lógica de negocio de la capa de presentación emplean este patrón, ejemplos de estas son: *UserController*, *ArtistController*, *MusicController*.

3.3.2 Patrones GoF

Los patrones GoF se pueden clasificar en varias categorías de acuerdo a un propósito específico: creacionales, estructurales y de comportamiento. A continuación, se presenta el patrón de este tipo utilizado en el diseño del sistema.

3.3.2.1 Creacionales

Factory Method (Método de fábrica): Define una interfaz para la creación de un objeto, pero permitiendo a las subclasses decidir de que clase instanciarlo. Permite, por tanto que una clase difiera de la instanciación en favor de sus subclasses.

Se utiliza en el componente Form de symfony, resolviendo los campos de los formularios implementados desde el servidor mediante herencia.

```
namespace Symfony\Component\Form;

class ResolvedFormTypeFactory
    implements ResolvedFormTypeFactoryInterface
{
    public function createResolvedType(
        FormTypeInterface $type,
        array $typeExtensions,
        ResolvedFormTypeInterface $parent = null
    )
    {
        return new ResolvedFormType(
            $type,
            $typeExtensions,
            $parent
        );
    }
}

$f = new ResolvedFormTypeFactory();
$form = $f->createResolvedType(new FormType());
$date = $f->createResolvedType(new DateType(), [], $form);
$bday = $f->createResolvedType(new BirthdayType(), [], $date);
```

Ilustración 9: Patrón GOF Fábrica abstracta.

Singleton: La utilización de este patrón asegura que cada clase tenga una instancia única la cual podrá ser accedida desde cualquier parte del sistema. En el controlador frontal del framework se puede observar el patrón en cuestión. Dentro del núcleo del mismo se puede observar su uso mediante `getInstance()`, `getDoctrine()` y `getEntityManager()`.

Ilustración 10: Patrón GOF Singleton.

3.3.2.2 Estructurales

Decorador: Adiciona responsabilidades a las clases de manera dinámica. Se utiliza en el componente `HttpKernel` para adicionar una capa de cache `Http`.

```
$dispatcher = new EventDispatcher();
$resolver = new ControllerResolver();
$store = new Store(__DIR__ . '/http_cache');

$httpKernel = new HttpKernel($dispatcher, $resolver);
$httpKernel = new HttpCache($httpKernel, $store);

$httpKernel
    ->handle(Request::createFromGlobals())
    ->send()
;
```

Ilustración 11: Patrón GOF Decorador.

3.3.2.3 Comportamiento

Iterativo: Se utilizó para atravesar el contenedor de objetos de Doctrine, filtrar colecciones de objetos, leer el contenido de un directorio recursivamente.

```
interface Iterator
{
    public function current();
    public function next();
    public function rewind();
    public function valid();
    public function key();
}
```

Ilustración 12: Patrón GOF Iterativo.

3.4 Diseño de Clases de los módulos

A continuación se muestra un fragmento de los diseños de clases de los módulos de la aplicación web Clave Cubana. Aquí se muestran las clases de forma amplia, debido a que en las tarjetas CRC se especifican las responsabilidades de cada clase.



Ilustración 13: Fragmento del diseño de clases del módulo seguridad.

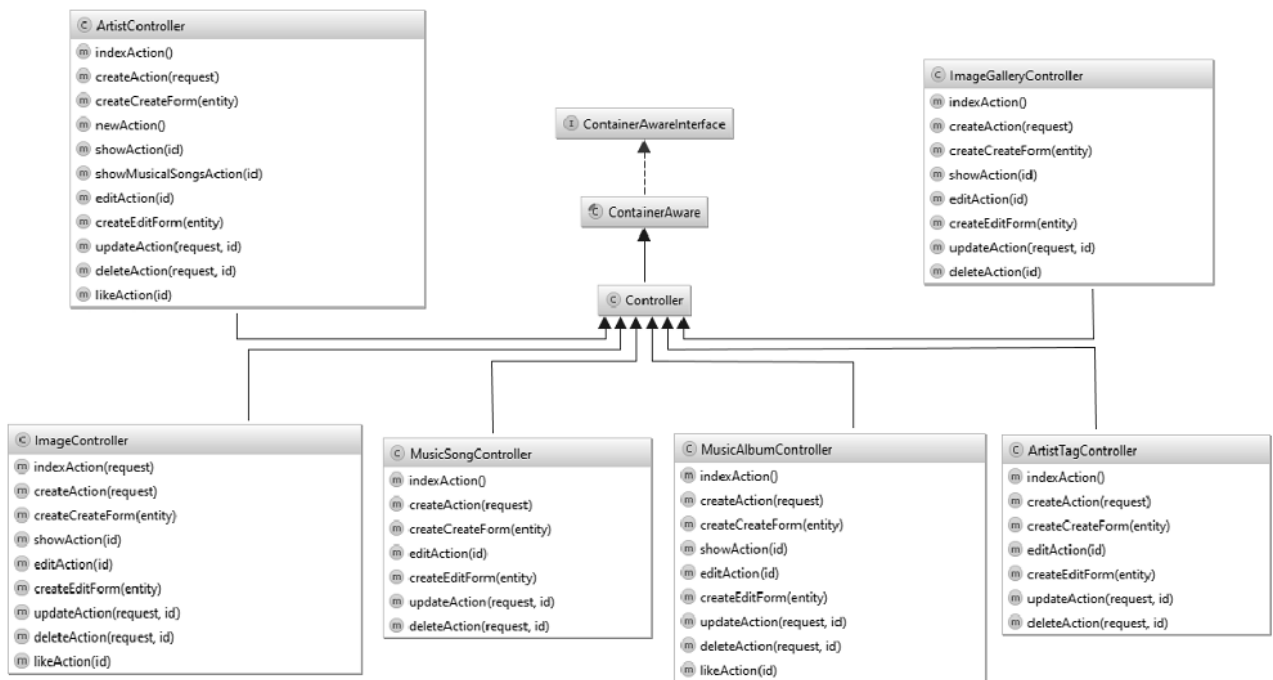


Ilustración 14: Fragmento del diseño de clases del módulo gestión.

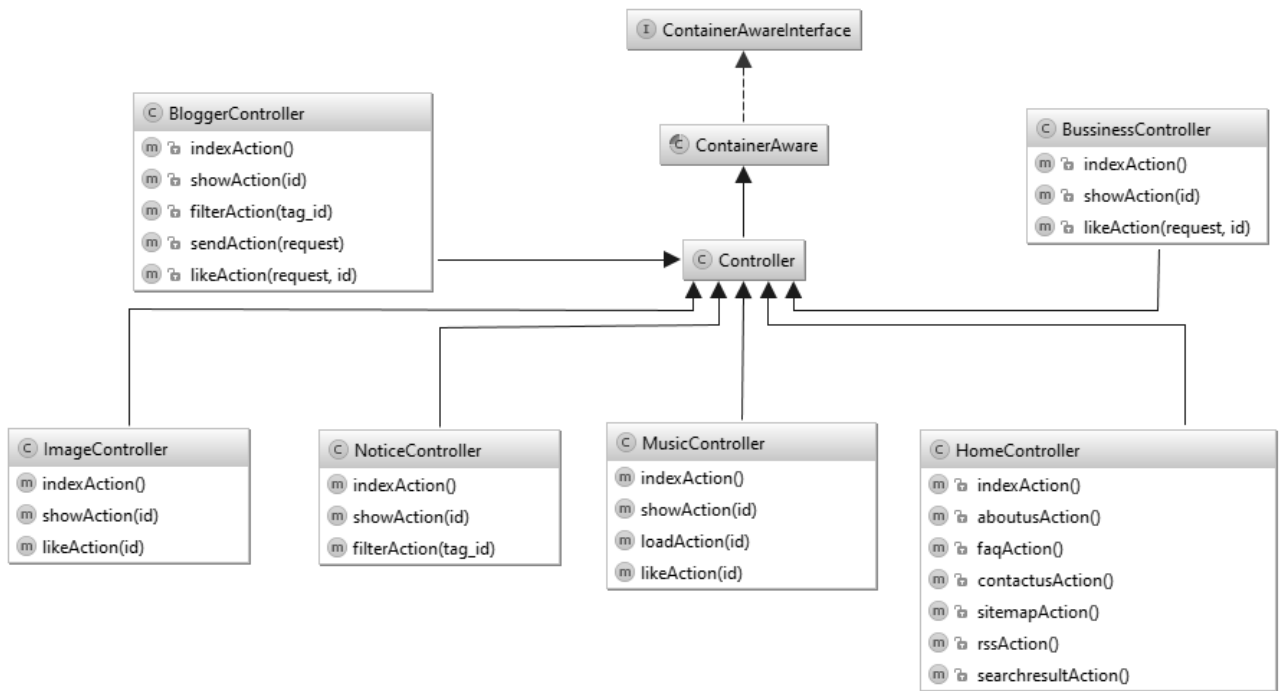


Ilustración 15: Fragmento del diseño de clases del módulo portal.

3.5 Tarjetas CRC

Las Tarjetas Clase-Responsabilidad-Colaborador (CRC), constituyen un modelo simple de organizar las clases más relevantes para las funcionalidades del sistema. Se utilizaron con el objetivo de desarrollar una representación organizada de las clases. Se encuentran divididas en tres secciones, correspondientes a sus siglas. A continuación, se describe la función que realizan:

Clase: Objeto que se analiza.

Responsabilidad: Cualquier función que la clase sabe o realiza.

Colaborador: Conjunto de clases requeridas para que una clase reciba la información necesaria con el objetivo de completar una responsabilidad.

A continuación, se muestran algunas de las Tarjetas CRC correspondientes al módulo Seguridad, el resto de las tarjetas se pueden observar en el [Anexo II](#).

Clase RolController	
Descripción: Se encarga de gestionar las acciones referentes al manejo de los roles de usuario.	
Responsabilidades	Colaboradores
Listar todos los elementos pertenecientes a la entidad " <i>tb_rol</i> ".	RolEntity, RolType
Verificar la validez de los formularios de creación de roles y almacenar el objeto <i>Rol</i> en base de dato.	RolEntity
Crear formulario de nuevo <i>Rol</i> .	RolEntity, RolType
Obtener el <i>Rol</i> a modificar y se invoca la acción de creación de formulario de modificación de <i>Rol</i> .	RolEntity
Crear el formulario de modificar <i>Rol</i> .	RolEntity, RolType
Verificar la validez de los formularios de modificación de roles y almacenar el objeto <i>Rol</i> modificado en base de dato.	RolEntity, RolType
Eliminar objeto <i>Rol</i> .	RolEntity

Tabla 6: Tarjeta CRC de la clase *RolController*.

Clase UserController	
Descripción: Se encarga de gestionar las acciones referentes al manejo de los usuarios del sistema.	
Responsabilidades	Colaboradores
Listar todos los elementos pertenecientes a la entidad "tb_user".	UserEntity, UserType
Verificar la validez de los formularios de creación de usuarios y almacenar el objeto <i>User</i> en base de dato.	UserEntity
Crear formulario de nuevo <i>User</i> .	UserEntity, UserType
Obtener el <i>User</i> a modificar y se invoca la acción de creación de formulario de modificación de <i>User</i> .	UserEntity
Crear el formulario de modificar <i>User</i> .	UserEntity, UserType
Verificar la validez de los formularios de modificación de usuarios y almacenar el objeto <i>User</i> modificado en base de dato.	UserEntity, UserType
Eliminar objeto <i>User</i> .	UserEntity

Tabla 7: Tarjeta CRC de la clase *UserController*.

Clase FrontendController	
Descripción: Se encarga de manejar las acciones referentes al acceso del usuario en el sistema.	
Responsabilidades	Colaboradores
Cargar la vista de autenticación del sistema.	ContactInfoRepository, AboutUsRepository, IconTipRepository, SocialNetworkRepository, BannerRepository
Cargar la vista para realizar acciones sobre los perfiles de usuarios.	ContactInfoRepository, AboutUsRepository, IconTipRepository, SocialNetworkRepository.
Cargar la vista para registrar un usuario en el sistema.	ContactInfoRepository, AboutUsRepository, IconTipRepository, SocialNetworkRepository
Cargar la vista referente al olvido de contraseña.	ContactInfoRepository, AboutUsRepository, IconTipRepository, SocialNetworkRepository
Cargar la vista para recuperación de contraseña.	ContactInfoRepository, AboutUsRepository, IconTipRepository, SocialNetworkRepository
Cargar la vista para el cambio de perfil de un usuario.	ContactInfoRepository, AboutUsRepository, IconTipRepository, SocialNetworkRepository

***Tabla 8:** Tarjeta CRC de la clase RepositoryController.*

3.6 Tareas de Ingeniería

La metodología XP, propone que una vez que se realice una HU es necesario probar y mostrar al cliente para comprobar si cumple con las especificaciones hechas por el mismo. Para esto, en el proceso de implementación, las HU fueron desglosadas en tareas de ingeniería (TI). Dichas TI se muestran mediante tablas que contienen un conjunto de campos, en los que se especifican los siguientes datos:

Número de la tarea: Los números deben ser consecutivos.

Número de HU: Número de la historia de usuario a la que pertenece la tarea.

Diseño e implementación de la aplicación web propuesta

Nombre Tarea: Nombre que identifica a la tarea.

Tipo de Tarea: Las tareas pueden ser de: Desarrollo, Corrección, Mejora, Otra (Especificar).

Puntos Estimados: Tiempo estimado en días que se le asignará a su desarrollo.

Fecha Inicio: Fecha en que se inicia el desarrollo de la tarea.

Fecha Fin: Fecha en que finaliza el desarrollo de la tarea.

Programador Responsable: Nombre y apellidos del programador.

Descripción: Breve descripción de la tarea.

A continuación, se muestran las TI correspondientes a las historias de usuarios de la primera iteración, el resto de las TI se pueden observar en el [Anexo III](#).

Tarea de Ingeniería	
Número de la tarea: 1	Número de la Historia de Usuario: 1
Nombre de la tarea: Listar galerías de imágenes.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,2
Fecha inicio : 28/1/2016	Fecha Fin: 28/1/2016
Descripción: Cuando el usuario selecciona del menú Multimedia , el submenú Imágenes , la opción Galerías , se ejecuta en la clase controladora <i>ImageGalleryControler</i> una acción que extrae todas las galerías guardadas en base de datos para mostrar de manera tabular el artista, la cantidad de imágenes, el nombre y si está visible o no.	

Tabla 9: TI 1. Listar galerías de imágenes.

Tarea de Ingeniería	
Número de la tarea: 2	Número de la Historia de Usuario: 1
Nombre de la tarea: Adicionar galería de imágenes.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,4
Fecha inicio : 29/1/2016	Fecha Fin: 1/2/2016
Descripción: Si se desea adicionar una nueva galería en el sistema, debe introducir el nombre y el artista en un formulario, luego son enviado a la Clase <i>ImageGalleryControler</i> para procesar los datos y almacenarlos en base de datos.	

Tabla 10: TI 2. Adicionar galería de imágenes.

Diseño e implementación de la aplicación web propuesta

Tarea de Ingeniería	
Número de la tarea: 3	Número de la Historia de Usuario: 1
Nombre de la tarea: Modificar galería de imágenes.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,4
Fecha inicio : 2/2/2016	Fecha Fin: 3/2/2016
Descripción: Si el usuario desea modificar una galería ya existente en el sistema, debe seleccionar la misma, seleccionar la opción de <i>Modificar</i> y luego cambiar el nombre y el artista, una vez aceptados son enviados nuevamente a la clase controladora para su posterior almacenamiento.	

Tabla 11: TI 3. Modificar galería de imágenes.

Tarea de Ingeniería	
Número de la tarea: 4	Número de la Historia de Usuario: 1
Nombre de la tarea: Eliminar galería de imágenes.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,2
Fecha inicio : 4/2/2016	Fecha Fin: 4/2/2016
Descripción: En caso de que se desee eliminar una galería el usuario debe seleccionarla, presionar la opción <i>Eliminar</i> y se mostrará un mensaje de confirmación una vez aceptado, los datos de la galería se enviarán a la clase controladora, los cuales serán buscados en la base de datos para su posterior eliminación.	

Tabla 12: TI 4. Eliminar galería de imágenes.

Tarea de Ingeniería	
Número de la tarea: 5	Número de la Historia de Usuario: 1
Nombre de la tarea: Mostrar galería de imágenes.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,2
Fecha inicio : 5/2/2016	Fecha Fin: 5/2/2016
Descripción: Si el usuario desea visualizar una galería de imágenes ya existente en el sistema, debe seleccionar la misma, seleccionar la opción de <i>Mostrar</i> , esta petición se envía a la clase	

Diseño e implementación de la aplicación web propuesta

controladora que extrae la galería especificada de la base de datos y luego la muestra en una nueva vista.

Tabla 13: TI 5. Mostrar galería de imágenes.

Tarea de Ingeniería	
Número de la tarea: 6	Número de la Historia de Usuario: 2
Nombre de la tarea: Listar catálogo de artistas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,2
Fecha inicio : 8/2/2016	Fecha Fin: 8/2/2016
Descripción: Cuando el usuario selecciona del menú Catálogo , el submenú Artistas , se ejecuta en la clase controladora una acción que extrae todos los catálogos guardados en base de datos para mostrar de manera tabular el nombre del catálogo, nombre artístico, categoría, si es visible o no y la cantidad de like que tiene.	

Tabla 14: TI 6. Listar catálogo de artistas.

Tarea de Ingeniería	
Número de la tarea: 7	Número de la Historia de Usuario: 2
Nombre de la tarea: Adicionar catálogo de artistas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,4
Fecha inicio : 9/2/2016	Fecha Fin: 10/2/2016
Descripción: Si se desea adicionar un nuevo catálogo, en el sistema, debe introducir en un formulario el nombre artístico, nombre del catálogo, si es visible o no y la categoría, luego son enviados a la clase <i>ArtistController</i> para procesar los datos y almacenarlos en base de datos.	

Tabla 15: TI 7. Adicionar catálogo de artistas.

Tarea de Ingeniería	
Número de la tarea: 8	Número de la Historia de Usuario: 2
Nombre de la tarea: Modificar catálogo de artistas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,4
Fecha inicio : 11/3/2016	Fecha Fin: 12/2/2016

Diseño e implementación de la aplicación web propuesta

Descripción: Si el usuario desea modificar un catálogo ya existente en el sistema, debe seleccionar el mismo, seleccionar la opción de *Modificar* y luego cambiar los datos, una vez aceptados son enviados nuevamente a la clase controladora para su posterior almacenamiento.

Tabla 16: TI 8. Modificar catálogo de artistas.

Tarea de Ingeniería	
Número de la tarea: 9	Número de la Historia de Usuario: 2
Nombre de la tarea: Eliminar catálogo de artistas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,2
Fecha inicio : 15/2/2016	Fecha Fin: 15/2/2016
Descripción: En caso de que se desee eliminar un catálogo, el usuario debe seleccionarlo, presionar la opción <i>Eliminar</i> y se mostrará un mensaje de confirmación, una vez aceptado, los datos del artista se enviarán a la clase controladora, los cuales serán buscados en la base de datos para su posterior eliminación.	

Tabla 17: TI 9. Eliminar catálogo de artistas.

Tarea de Ingeniería	
Número de la tarea: 10	Número de la Historia de Usuario: 2
Nombre de la tarea: Mostrar catálogo de artistas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0,2
Fecha inicio : 16/2/2016	Fecha Fin: 16/2/2016
Descripción: Si el usuario desea visualizar un catálogo ya existente en el sistema, debe seleccionar el mismo, seleccionar la opción de <i>Mostrar</i> , esta petición se envía a la clase controladora que extrae el artista específico de la base de datos y luego lo muestra en una nueva vista.	

Tabla 18: TI 10. Mostrar catálogo de artistas.

3.7 Estándares de Codificación

Un estándar de codificación agrupa los aspectos relacionados con la generación del código. Los programadores lo realizan para trabajar de forma coordinada dentro del proyecto, de acuerdo con el tipo de lenguaje que utilizan y las normas que propone el mismo, tratando de lograr un código claro y legible tanto para él como para otro programador. (Microsoft Developer Network, 2012)

A continuación, se muestran algunos estándares de codificación que se tuvieron en cuenta para la implementación de los módulos de Clave Cubana:

- ❖ **Indentación:** Se emplearon cuatro espacios como unidad de indentación. Ejemplo:

```
... /**
... *
... * @return \Symfony\Component\HttpFoundation\Response
... */
... public function logAction(){
...     $locale = $this->get('request')->getLocale();
...     $em = $this->getDoctrine()->getEntityManager();
...     $contact = $em->getRepository('AdminClaveBundle:ContactInfo')->find(1);
...     $aboutus = $em->getRepository('AdminClaveBundle:AboutUs')->find(1);
...     $icons = $em->getRepository('AdminClaveBundle:IconTip')->findByState(1);
...     $social = $em->getRepository('AdminClaveBundle:SocialNetwork')->findByState(1, 0);
...     $favicon = $em->getRepository('AdminClaveBundle:Banner')->findByType(1, 'icon');
...     $logo = $em->getRepository('AdminClaveBundle:Banner')->findByType(1, 'logo');
...     return $this->render('AuthBundle:Frontend:login.html.twig',
...         array(
...             'logo' => $logo,
...             'locale' => $locale,
...             'favicon' => $favicon,
...             'aboutus' => $aboutus,
...             'contact' => $contact,
...             'social' => $social,
...             'icons' => $icons,
...         )
...     );
... }
```

Ilustración 16: Estándar de codificación Indentación.

- ❖ **Longitud de línea:** Se evitó la utilización de más de 80 caracteres en una línea.
- ❖ **Comentarios:** Se utilizaron los comentarios limitados por “ */* comentario */*” para mostrar información que comprende más de una línea. También se utilizaron los comentarios de la siguiente forma “*//*”, para separar un conjunto de métodos por una sola línea. Ejemplos:

```
... // Slide.controller.
...
... /**
... * Lists all Slide entities.
... *
... * @Route("/", name="slide")
... * @Method("GET")
... * @Template()
... */
```

Ilustración 17: Estándar de codificación Comentarios.

- ❖ **Declaraciones:** Los métodos y variables se declararon con nombres asociados a la función por la cual fueron creados. En caso de nombres compuestos, el primero se definió en minúscula y los demás comenzando con mayúscula. En caso de separarlos no se utilizó espacios, sino guiones bajos (<nombre1>_<nombre2>). Las clases a diferencia de las variables y los métodos se declararon con la inicial del primer nombre en mayúscula.

```
../**
...*. @var integer
...*
...*. @ORM\Column(name="id", type="integer")
...*. @ORM\Id
...*. @ORM\GeneratedValue(strategy="IDENTITY")
...*/
..protected $id;
```

Ilustración 18: Estándar de codificación Declaración de variables.

```
..public function setWidth($width)
..{
...$this->width = $width;
...return $this;
..}
```

Ilustración 19: Estándar de codificación Declaración de métodos.

```
<?php
namespace System\AdminClaveBundle;
use Symfony\Component\HttpKernel\Bundle\Bundle;
class AdminClaveBundle extends Bundle
{
}
```

Ilustración 20: Estándar de codificación Declaración de clases.

- ❖ **Sentencia if, try/ catch:** Se definió para estos dos tipos de sentencias la siguiente estructura:

```
try {
    if (!$this->get('security.context')->isGranted('ROLE_ADMIN'))
        return $this->render(':backend:500.html.twig', array('message' => 'Access denied.));
    else {
        $entity = new Slide();
        $form = $this->createForm($entity);
        $form->handleRequest($request);
        if ($form->isValid()) {
            $em = $this->getDoctrine()->getManager();
            $entity->setImage("");
            $files = $request->files->get("system_adminclavebundle_slide");
            if (!empty($files)) {
                if (isset($files["image"])) {
                    $host_name = ((strstr(strtolower($_SERVER['SERVER_PROTOCOL']), 'https')) ? 'https' : 'http') . "://";
                    $files["image"]->move("uploads/site/slider/", $files["image"]->getClientOriginalName());
                    $entity->setImage($host_name . "/uploads/site/slider/" . $files["image"]->getClientOriginalName());
                }
            }
            $this->get('activity_controller')->registerAction($em, $this->get('security.context')->getToken()->getUser()->getName(), $entity);
            $em->persist($entity);
            $em->flush();
            return $this->redirect($this->generateUrl('slide'));
        }
        return array('entity' => $entity, 'form' => $form->createView(),);
    }
} catch (\Exception $ex) {
    return $this->render(':backend:500.html.twig', array('message' => 'Sorry, we are having a problem. Please contact to...));
}
```

Ilustración 21: Estándar de codificación de sentencias *if* y *try/catch*.

En el *if* y el *try/catch*, se utilizó llaves para abrir y cerrar el bloque de acciones que se ejecutan dadas condiciones.

3.8 Conclusiones del capítulo

Con la realización de este capítulo se determinó el flujo de actividades que se lleva a cabo en la aplicación web Clave Cubana. Con el empleo de los patrones de diseño y los estándares de codificación se logró que los programadores realizaran un trabajo coordinado y estructurado. Haciendo uso de las tarjetas CRC se describen las clases del sistema, y las funcionalidades de las HU mediante las TI.

Capítulo 4: Pruebas de la aplicación web propuesta

Introducción

La fase de prueba es una de las actividades más importantes dentro del desarrollo de software, porque mide la calidad y el éxito de las funcionalidades que se van desarrollando. En este tipo de actividad el cliente puede evaluar los requerimientos establecidos anteriormente con los productos funcionales que se van obteniendo.

4.1 Estrategia de Pruebas

La metodología XP propone realizar tantas pruebas como sea posible con el objetivo de disminuir los errores que puedan afectar los resultados. Se encuentran divididas en dos grupos, pruebas unitarias y pruebas de aceptación.

Pruebas Unitarias: Son consideradas una de las etapas imprescindibles en XP. Este tipo de prueba se debe realizar antes de escribir el código, de forma tal que se les aplica a todos los métodos que se implementen una vez terminados y antes de ser publicados. Que todo el código realizado pase correctamente las pruebas unitarias es lo que garantiza que funcionen en conjunto con los demás. En este sentido, las pruebas deben ser guardadas en conjunto con el sistema, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo (Comunidad digital del conocimiento, 2012).

Pruebas de Aceptación: Son creadas en base a las historias de usuario, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como Pruebas de Caja Negra¹⁶. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información (Comunidad digital del conocimiento, 2012).

¹⁶ Pruebas de Caja Negra: Permiten obtener un conjunto de condiciones de entrada que ejerciten todos los requisitos funcionales de un software.

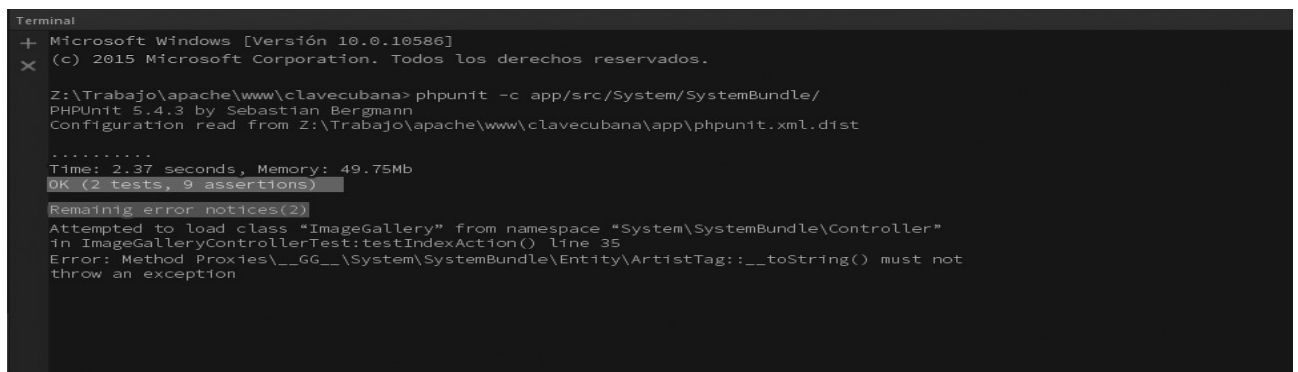
4.1.1 Pruebas Unitarias

Para aplicar las pruebas unitarias a las funcionalidades se utilizó PHPUnit¹⁷. Estas pruebas fueron llevadas a cabo por los desarrolladores, encargados de verificar el código de forma automática, se aplicaron a todas las funcionalidades implementadas en los módulos comprobando el funcionamiento de cada una de estas. Las pruebas unitarias se fueron realizando al terminar cada una de las siete iteraciones en las que se desarrollaron las funcionalidades de la aplicación web.

A continuación, se muestran los resultados de las pruebas en cada iteración:

Iteración Uno:

En esta iteración se comprueba el funcionamiento de las funcionalidades: gestionar galería de imágenes y gestionar catálogo de artistas. Como se muestra en la ilustración 22, la primera prueba unitaria aplicada arrojó como resultado dos no conformidades. Estas no conformidades fueron solucionadas y se realizó nuevamente la prueba unitaria, dando como resultado cero no conformidades como se puede observar en la ilustración 23.



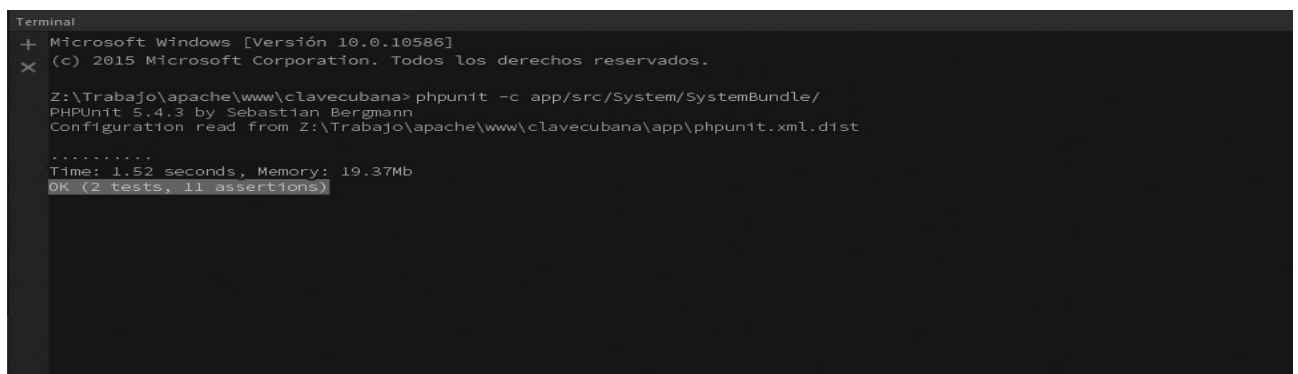
```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
X (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist

.....
Time: 2.37 seconds, Memory: 49.75Mb
OK (2 tests, 9 assertions)

Remaining error notices(2)
Attempted to load class "ImageGallery" from namespace "System\SystemBundle\Controller"
in ImageGalleryControllerTest::testIndexAction() line 35
Error: Method Proxies\__GG__\System\SystemBundle\Entity\ArtistTag::__toString() must not
throw an exception
```

Ilustración 22: Primera prueba unitaria aplicada a las funcionalidades de la iteración uno.



```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
X (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist

.....
Time: 1.52 seconds, Memory: 19.37Mb
OK (2 tests, 11 assertions)
```

Ilustración 23: Segunda prueba unitaria aplicada a las funcionalidades de la iteración uno.

Iteración Dos:

¹⁷ PHPUnit: Framework de desarrollo con PHP, orientado a pruebas de código. Utiliza aserciones para verificar que el comportamiento de una unidad de código es el esperado.

Pruebas de la aplicación web propuesta

En esta iteración se comprueba el funcionamiento de las funcionalidades: gestionar álbum musical y gestionar galerías de videos. Como se muestra en la ilustración 24, la primera prueba unitaria aplicada en esta iteración arrojó como resultado una no conformidad. Esta no conformidad fue solucionada y se realizó nuevamente la prueba unitaria, dando como resultado cero no conformidades como se puede observar en la ilustración 25.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
x (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist

.....
Time: 2.37 seconds, Memory: 49.75Mb
OK (2 tests, 10 assertions)

Remaining deprecation notices(1)
getEntityManager is deprecated since Symfony 2.1. Use getManager instead: 1x
in MusicControllerTest::testShowAction from System\SystemBundle\Test\MusicController
```

Ilustración 24: Primera prueba unitaria aplicada a las funcionalidades de la iteración dos.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
x (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist

.....
Time: 1.21 seconds, Memory: 19.87Mb
OK (2 tests, 11 assertions)
```

Ilustración 25: Segunda prueba unitaria aplicada a las funcionalidades de la iteración dos.

Iteración Tres:

En esta iteración se comprueba el funcionamiento de las funcionalidades: gestionar cargador de archivos y gestionar redes sociales. Como se puede observar en la ilustración 26, al aplicar una prueba unitaria en esta iteración no se obtuvo no conformidades, por lo que las funcionalidades están funcionalmente correctas.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
x (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist
.....
Time: 2.01 seconds, Memory: 18.42Mb
OK (2 tests, 11 assertions)
```

Ilustración 26: Primera prueba unitaria aplicada a las funcionalidades de la iteración tres.

Iteración Cuatro:

En esta iteración se comprueba el funcionamiento de las funcionalidades: gestionar lenguajes y gestionar perfil de usuario. Como se puede observar en la ilustración 27, al aplicar una prueba unitaria en esta iteración no se obtuvo no conformidades, por lo que las funcionalidades están funcionalmente correctas.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
x (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist
.....
Time: 2.38 seconds, Memory: 24.13Mb
OK (2 tests, 11 assertions)
```

Ilustración 27: Primera prueba unitaria aplicada a las funcionalidades de la iteración cuatro.

Iteración Cinco:

En esta iteración se comprueba el funcionamiento de las funcionalidades: gestionar usuarios con acceso al sistema, gestionar categorías de artistas, gestionar roles de usuarios. Como se puede observar en la ilustración 28, al aplicar una prueba unitaria en esta iteración no se obtuvo no conformidades, por lo que las funcionalidades están funcionalmente correctas.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
X (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist
.....
Time: 3.17 seconds, Memory: 30.07Mb
OK (3 tests, 17 assertions)
```

Ilustración 28: Primera prueba unitaria aplicada a las funcionalidades de la iteración cinco.

Iteración Seis:

En esta iteración se comprueba el funcionamiento de las funcionalidades: gestionar noticias, gestionar categorías de noticias, gestionar artículos del blog y gestionar categorías de los artículos del blog. Como se muestra en la ilustración 29, la primera prueba unitaria aplicada en esta iteración arrojó como resultado tres no conformidades. Estas no conformidades fueron solucionadas y se realizó nuevamente la prueba unitaria, dando como resultado cero no conformidades como se puede observar en la ilustración 30.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
X (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist
.....
Time: 3.54 seconds, Memory: 63.25Mb
OK (4 tests, 21 assertions)

Remaining error notices(1) deprecation(2)
getEntityManager is deprecated since Symfony 2.1. Use getManager instead: 2x
  1x in NoticeControllerTest::testShowAction from System\SystemBundle\Test\NoticeController
  1x in BlogItemCategoryControllerTest::testIndexAction from System\BloggerBundle\Test\BloggerController
The controller must return a response (null given). Did you forgot to add a return statement somewhere
in your controller?
```

Ilustración 29: Primera prueba unitaria aplicada a las funcionalidades de la iteración seis.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
X (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist

.....
Time: 4.02 seconds, Memory: 69.07Mb
OK (4 tests, 22 assertions)
```

Ilustración 30: Primera prueba unitaria aplicada a las funcionalidades de la iteración seis.

Iteración Siete:

En esta iteración se comprueba el funcionamiento de las funcionalidades: gestionar comentarios de los artículos del blog, gestionar logotipos y gestionar información de la empresa e información de contacto. Como se muestra en la ilustración 31, la primera prueba unitaria aplicada en esta iteración arrojó como resultado una no conformidad. Esta no conformidad fue solucionada y se realizó nuevamente la prueba unitaria, dando como resultado cero no conformidades como se puede observar en la ilustración 32.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
X (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist

.....
Time: 2.26 seconds, Memory: 33.32Mb
OK (3 tests, 16 assertions)

Remaining error notices(1)
Notice: Undefined variable: entities in IconTipControllerTest::testIndexAction()
in line 43
```

Ilustración 31: Primera prueba unitaria aplicada a las funcionalidades de la iteración siete.

```
Terminal
+ Microsoft Windows [Versión 10.0.10586]
X (c) 2015 Microsoft Corporation. Todos los derechos reservados.

Z:\Trabajo\apache\www\clavecubana> phpunit -c app/src/System/SystemBundle/
PHPUnit 5.4.3 by Sebastian Bergmann
Configuration read from Z:\Trabajo\apache\www\clavecubana\app\phpunit.xml.dist

.....
Time: 2.26 seconds, Memory: 33.32Mb
OK (3 tests, 17 assertions)
```

Ilustración 32: Segunda prueba unitaria aplicada a las funcionalidades de la iteración siete.

Como se muestra en la ilustración 33, en total se encontraron siete no conformidades, sin embargo, se resolvieron en las iteraciones que se encontraron.



Ilustración 33: No conformidades encontradas al realizar las pruebas unitarias por iteración.

4.1.2 Pruebas de Aceptación

Las pruebas de aceptación fueron creadas por los clientes en base a las HU implementadas. En ellas, el cliente se enfocó en las características generales y las funcionalidades del sistema, creando diversos escenarios para comprobar que una HU fue desarrollada correctamente.

Se realizaron siete iteraciones en estas pruebas, y en ninguna de ellas se obtuvo resultado insatisfactorio. A continuación, se muestra un escenario para los casos de pruebas de aceptación realizados para la iteración uno. El resto de los casos de prueba se muestran en el [Anexo V](#).

Caso de Prueba de Aceptación	
Código: HU1_CP1	Historia de Usuario: 1
Nombre: Insertar datos correctos al adicionar una galería de imágenes.	
Descripción: Se desea probar que el sistema permite al usuario (administrador) introducir los datos correctos correspondientes a la inserción de una nueva galería de imágenes.	
Condiciones de Ejecución: Conjunto de datos válidos de la galería de imágenes.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Multimedia</i>.❖ El administrador presiona la opción <i>Nueva Galería de Imágenes</i>.❖ El administrador introduce los datos requeridos para la creación de la nueva galería de imágenes.❖ El administrador presiona el botón <i>Aceptar</i>.	
Resultado Esperado: Se muestra el listado de galerías de imágenes y se puede observar la nueva galería.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 19: Caso de prueba de aceptación HU1_CP1.

Caso de Prueba de Aceptación	
Código: HU1_CP2	Historia de Usuario: 1
Nombre: Insertar datos erróneos al adicionar una galería de imágenes.	
Descripción: Se desea probar que el sistema no permite al usuario (administrador) introducir los datos correctos correspondientes a la inserción de nueva galería de imágenes.	
Condiciones de Ejecución: Conjunto de datos incorrectos o incompletos de la galería de imágenes.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Multimedia</i>.❖ El administrador presiona la opción <i>Nueva Galería de Imágenes</i>.❖ El administrador introduce los datos requeridos para la creación de la nueva galería de imágenes.❖ El administrador presiona el botón <i>Aceptar</i>.	
Resultado Esperado: El sistema mostrará una notificación al administrador acerca de los datos erróneos o incompletos para que vuelva a introducirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 20: Caso de prueba de aceptación HU1_CP1.

Caso de Prueba de Aceptación	
Código: HU1_CP3	Historia de Usuario: 1
Nombre: Eliminar galería de imágenes.	
Descripción: Se desea probar que el sistema permita al usuario (administrador) eliminar la galería de imágenes seleccionada.	
Condiciones de Ejecución: Debe existir al menos una galería para que pueda ser borrada.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Multimedia</i>.❖ El administrador presiona el link <i>Imágenes</i>.❖ El administrador presiona el link <i>Galerías</i>.❖ El administrador selecciona la galería a eliminar.❖ El administrador presiona el botón eliminar.❖ El sistema muestra un cartel que dice “Advertencia: Realmente desea eliminar el elemento seleccionado junto con todo su contenido”.❖ El administrador presiona el botón de aceptar.	
Resultado Esperado: El sistema mostrará la lista de galerías, y no aparecerá la galería eliminada.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 21: Caso de prueba de aceptación HU1_CP3.

Caso de Prueba de Aceptación	
Código: HU1_CP4	Historia de Usuario: 1
Nombre: Modificar galería de imágenes introduciendo datos correctos.	
Descripción: Se desea probar que el sistema permita al usuario (administrador) modificar la galería de imágenes seleccionada.	
Condiciones de Ejecución: Debe existir al menos una galería para que pueda ser modificada.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Multimedia</i>.❖ El administrador presiona el link <i>Imágenes</i>.❖ El administrador presiona el link <i>Galerías</i>.❖ El administrador selecciona la galería a modificar.❖ El administrador presiona el botón modificar.❖ El sistema muestra un formulario para que el usuario introduzca los datos.❖ El administrador introduce los datos correctos.❖ El administrador presiona el botón de aceptar.	
Resultado Esperado: El sistema mostrará la lista de galerías, y aparecerá la galería modificada.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 22: Caso de prueba de aceptación HU1_CP4.

Caso de Prueba de Aceptación	
Código: HU1_CP5	Historia de Usuario: 1
Nombre: Modificar galería de imágenes introduciendo datos incorrectos.	
Descripción: Se desea probar que el sistema no permita al usuario (administrador) modificar la galería de imágenes seleccionada.	
Condiciones de Ejecución: Debe existir al menos una galería para que pueda ser modificada.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Multimedia</i>.❖ El administrador presiona el link <i>Imágenes</i>.❖ El administrador presiona el link <i>Galerías</i>.❖ El administrador selecciona la galería a modificar.❖ El administrador presiona el botón modificar.❖ El sistema muestra un formulario para que el usuario introduzca los datos.❖ El administrador introduce los datos incorrectos.❖ El administrador presiona el botón de aceptar.	
Resultado Esperado: El sistema especificará los campos que contengan datos erróneos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 23: Caso de prueba de aceptación HU1_CP5.

Caso de Prueba de Aceptación	
Código: HU2_CP1	Historia de Usuario: 2
Nombre: Insertar datos correctos al adicionar un catálogo de artistas.	
Descripción: Se desea probar que el sistema permite al usuario (administrador) introducir los datos correctos correspondientes a la inserción de un nuevo catálogo de imágenes.	
Condiciones de Ejecución: Conjunto de datos válidos del catálogo de artistas.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Catálogo</i>.❖ El administrador presiona la opción <i>Nuevo catálogo de artistas</i>.❖ El administrador introduce los datos requeridos para la creación del nuevo catálogo de artistas.❖ El administrador presiona el botón <i>Aceptar</i>.	
Resultado Esperado: Se muestra el listado de catálogos de artistas, y se puede observar el nuevo catálogo nuevo.	
Evaluación de la Prueba: Satisfactoria.	

*Tabla 24:*Caso de prueba de aceptación HU2_CP1.

Caso de Prueba de Aceptación	
Código: HU2_CP2	Historia de Usuario: 2
Nombre: Insertar datos erróneos al adicionar un catálogo de artistas.	
Descripción: Se desea probar que el sistema no permite al usuario (administrador) introducir los datos incorrectos correspondientes a la inserción de nueva galería de imágenes.	
Condiciones de Ejecución: Conjunto de datos incorrectos o incompletos del catálogo de artistas.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Catálogo</i>.❖ El administrador presiona la opción <i>Nuevo catálogo de artistas</i>.❖ El administrador introduce los datos erróneos requeridos para la creación del nuevo catálogo de artistas.❖ El administrador presiona el botón <i>Aceptar</i>.	
Resultado Esperado: El sistema mostrará una notificación al administrador acerca de los datos erróneos o incompletos para que vuelva a introducirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 25: Caso de prueba de aceptación HU2_CP2.

Caso de Prueba de Aceptación	
Código: HU2_CP3	Historia de Usuario: 2
Nombre: Eliminar catálogo de artistas.	
Descripción: Se desea probar que el sistema permita al usuario (administrador) eliminar el catálogo de artistas.	
Condiciones de Ejecución: Debe existir al menos un catálogo de artistas.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Catálogo</i>.❖ El administrador selecciona el catálogo a eliminar.❖ El administrador presiona la opción eliminar.❖ El sistema muestra un cartel que dice “Advertencia: Realmente desea eliminar el elemento seleccionado junto con todo su contenido”.❖ El administrador presiona el botón <i>Aceptar</i>.	
Resultado Esperado: El sistema mostrará la lista catálogos de artistas y no aparecerá el catálogo eliminado.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 26: Caso de prueba de aceptación HU2_CP3.


Caso de Prueba de Aceptación	
Código: HU2_CP4	Historia de Usuario: 2
Nombre: Modificar catálogo de artistas introduciendo datos correctos.	
Descripción: Se desea probar que el sistema permita al usuario (administrador) modificar el catálogo de artistas seleccionado.	
Condiciones de Ejecución: Debe existir al menos un catálogo para que pueda ser modificado.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Catálogo</i>.❖ El administrador selecciona el catálogo a modificar.❖ El administrador presiona el botón modificar.❖ El sistema muestra un formulario para que el usuario introduzca los datos.❖ El administrador introduce los datos correctos.❖ El administrador presiona el botón de aceptar.	
Resultado Esperado: El sistema mostrará la lista de catálogos, y aparecerá el catálogo modificado.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 27: Caso de prueba de aceptación HU2_CP4.

Caso de Prueba de Aceptación	
Código: HU2_CP5	Historia de Usuario: 2
Nombre: Modificar catálogo de artistas introduciendo datos incorrectos.	
Descripción: Se desea probar que el sistema no permita al usuario (administrador) modificar el catálogo de artistas seleccionado.	
Condiciones de Ejecución: Debe existir al menos un catálogo para que pueda ser modificado.	
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">❖ El administrador presiona el link <i>Catálogo</i>.❖ El administrador selecciona el catálogo a modificar.❖ El administrador presiona el botón modificar.❖ El sistema muestra un formulario para que el usuario introduzca los datos.❖ El administrador introduce los datos incorrectos.❖ El administrador presiona el botón de aceptar.	
Resultado Esperado: El sistema especificará los campos que contengan datos erróneos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 28: Caso de prueba de aceptación HU2_CP5.

4.2 Aval de aceptación del cliente

	Casa Matriz Sta. Ave. No. 8010 e/ 80 y 82, Playa, La Habana, Cuba. Tels.: (537) 204 2710 • Fax: (537) 204 2033 • E-mail: infoartex@artexvca PROMOCIONES ARTÍSTICAS Y LITERARIAS S.A. www.artexsa.com
Tels	


Opinión del Usuario del Trabajo de Diploma

El Trabajo de Diploma, titulado "**Aplicación web para la promoción y gestión de la información de los artistas de la agencia de representaciones artísticas Clave Cubana**" ha sido realizado para la Agencia de Representaciones Artísticas Clave Cubana perteneciente al grupo empresarial Artex S.A. Hasta la fecha actual y teniendo en cuenta la última revisión realizada del proyecto de desarrollo, se considera que el trabajo realizado satisface en un 90% de su propósito, atendiendo el compromiso de los Diplomantes de continuar el mismo hasta su finalización.

Los resultados de este Trabajo de Diploma le reportan a la Agencia de Representaciones Artísticas Clave Cubana y a Artex S.A. los beneficios siguientes:

- Se logra una herramienta que automatiza la promoción de artistas de Clave
- Se optimiza de manera considerable la gestión de la información de manera general de todos los artistas representados por clave.
- Se logra un portal web corporativo para Clave que representa la agencia en Internet.
- Facilita el manejo de la información referente a los artistas
- Se logra un sistema informatizado fácil de usar y que no requiere de mucha experiencia para su uso, presenta un ambiente amigable, sencillo y cumple con los requisitos de diseño especificados.
- Como resultado de las revisiones realizadas hasta el momento, el trabajo de diploma cumple con las necesidades propuestas por Clave y han sido revisadas por Artex.

Y para que conste así la conformidad de la Institución en lo antes expuesto, se firma la presente a los 6 días del mes de junio del año 2016.



Yisel Soriano Arias
Téc A. Planeamiento
Gerencia de Comunicación




Ilustración 34: Aval de aceptación del cliente.

4.3 Conclusiones del capítulo

En el presente capítulo se realizaron las pruebas a la aplicación web Clave Cubana, siguiendo las especificaciones de la cuarta fase de XP. A medida que se implementaron las funcionalidades se les realizaron las pruebas unitarias y de aceptación. Las no conformidades que se encontraron se solucionaron en las mismas iteraciones, de esta forma quedaron evaluadas las funcionalidades reflejadas en cada TI, logrando la obtención de un software con mayor calidad y menos defectos.

Conclusiones Generales

Una vez desarrollada la presente investigación, cumplidas las tareas de investigación y el objetivo general, se puede arribar a las siguientes conclusiones:

- ❖ Se definió el marco teórico investigativo donde se decidió utilizar las tecnologías Symfony, WAMP, JQuery, Ajax y Bootstrap, logrando soberanía tecnológica.
- ❖ Al realizar el estudio de las aplicaciones web que promocionan y gestionan información de artistas, se determinó un conjunto de funcionalidades que se añadieron a las especificadas por el cliente, para su implementación.
- ❖ Se obtuvo como resultado una aplicación web para promocionar y gestionar la información de los artistas de la agencia Clave Cubana.
- ❖ Con el empleo de patrón arquitectónico, patrones de diseño y estándares de codificación se realizó un trabajo organizado y coordinado a lo largo de la implementación del proyecto. Permitieron que la aplicación sea reutilizable, que las clases presenten alta cohesión y bajo acoplamiento.
- ❖ Se validó la aplicación web a haciendo uso de las pruebas unitarias y de aceptación, comprobándose la efectividad del código y la satisfacción del cliente con las funcionalidades presentadas.

Recomendaciones

Se recomienda para futuras investigaciones:

- ❖ Mantener actualizado la aplicación web con futuras versiones del framework que se liberen.
- ❖ Permitir a la aplicación integrarse con el servidor FTP institucional de Artex S.A, lo cual permitiría un mejor almacenamiento del contenido multimedia.

Referencias bibliográficas

Artex.S.A. ARTEX, donde lo cubano se hace arte. [En línea] ARTEX. [Citado el: 13 de 1 de 2016.] <http://www.artexsa.com/es/aboutus>. 3.

Comunidad digital del conocimiento. 2012. Las pruebas de integración del software. [En línea] 25 de 9 de 2012. [Citado el: 11 de 11 de 15.] <http://www.academica.mx/blogs/las-pruebas-integracion-software>.

Daniel Ernesto Aldás Mena, Alejandra Andrade Cadena. 2010. *Guí Práctica para el uso de Patrones de Diseño en el Desarrollo del Software*. Quito : s.n., 2010.

Díaz, Rafael de la Osa. 2012. Cubarte: El portal de la cultura cubana. [En línea] 26 de 04 de 2012. [Citado el: 15 de 10 de 2015.] <http://www.cubarte.com>. Portal Cubarte Columnas MITO Y REALIDAD Las TIC y las Industrias Culturales (Parte II).

Fuentes, Osay Gonzáles. 2012. La Comunidad UCI de PHP. *Curso para participantes en Symfony2*. [En línea] 31 de 5 de 2012. [Citado el: 13 de 11 de 2015.] https://php.uci.cu/articles.php?article_id=87.

García, Arailaisy Rosabal. 2011. Radio Cadena Agramonte. [En línea] Radio Camagüeyana, 9 de 2 de 2011. [Citado el: 2015 de 11 de 12.] <http://www.cadenagramonte.cu/articulos/ver/12669:pasos-solidos-en-camagueey-por-la-informatizacion-de-la-sociedad>. 2.

López, Eva Santana. 2003. *Las promociones publicitarias en la prensa. Evolución y análisis sistémico de las promociones realizadas entre los años 1990-1998 en la prensa de información general de Catalunya*. España, 2003.

Luis Calabria, Pablo Piriz. 2003. *Metodología XP*. Uruguay : s.n., 2003.

Martín, David Andrés. 2009. *Música Industrial y promoción. ¿Cómo ha cambiado el marketing musical?* 2009.

Martínez, Víctor Martínez. 2015. *LA PUBLICIDAD EN LA INDUSTRIA MUSICAL. LA COMUNICACIÓN Y LA PROMOCIÓN DE LA MÚSICA EN VIVO EN LAS*. 2015.

Mora, Sergio Luján. 2002. *Programación de Aplicaciones Web: Historia, Principios básicos y Cliente Web*. Alicante : Club Univeristario, 2002.

Morales, Flores Elusa. 2004. Bibliodocencia. [En línea] 2004. [Citado el: 4 de 6 de 2016.] http://www.bibliodocencia.com/4/4_6.pdf.

Moreno, Nancy González y González, Fermín O Rodríguez. 2002. *La gestión de la información como base a la gestión del conocimiento y del aprendizaje en las organizaciones en las universidades*. Habana : s.n., 2002.

Penadés, Patricio Letelier. 2010. *Metodologías ágiles para el desarrollo de software: Extrema Programming*. Valencia : s.n., 2010.

Potencier, F y Zaninotto, F. 2009. *Symfony 1.2, la guía definitiva*. 3 de 10 de 2009.

Referencias bibliográficas

Project Team Of JQuery. 2010. JQuery Project. [En línea] 2010. [Citado el: 03 de 11 de 2015.] <http://jquery.com>.

Ron Morrison, Flavio Oquendo. 2005. *Software Architecture*. Italia : Board, 2005.

Santesmases, Miguel Mestre. 1996. *Términos de marketing. Diccionario-Base de Datos*. Madrid : Ediciones Pirámide, 1996.

Bibliografía

Artex.S.A. ARTEX, donde lo cubano se hace arte. [En línea] ARTEX. [Citado el: 13 de 1 de 2016.] <http://www.artexsa.com/es/aboutus>. 3.

Comunidad digital del conocimiento. 2012. Las pruebas de integración del software. [En línea] 25 de 9 de 2012. [Citado el: 11 de 11 de 15.] <http://www.academica.mx/blogs/las-pruebas-integracion-software>.

Daniel Ernesto Aldás Mena, Alejandra Andrade Cadena. 2010. *Guí Práctica para el uso de Patrones de Diseño en el Desarrollo del Software*. Quito : s.n., 2010.

Departamento de Lenguajes y Sistemas Informáticos. 2014. *Patrones Arquitectónicos*. Sevilla : s.n., 2014.

Díaz, Rafael de la Osa. 2012. Cubarte: El portal de la cultura cubana. [En línea] 26 de 04 de 2012. [Citado el: 15 de 10 de 2015.] <http://www.cubarte.com>. Portal Cubarte Columnas MITO Y REALIDAD Las TIC y las Industrias Culturales (Parte II).

Fuentes, Osay Gonzáles. 2012. La Comunidad UCI de PHP. *Curso para participantes en Symfony2*. [En línea] 31 de 5 de 2012. [Citado el: 13 de 11 de 2015.] https://php.uci.cu/articles.php?article_id=87.

García, Arailaisy Rosabal. 2011. Radio Cadena Agramonte. [En línea] Radio Camagüeyana, 9 de 2 de 2011. [Citado el: 2015 de 11 de 12.] <http://www.cadenagramonte.cu/articulos/ver/12669:pasos-solidos-en-camagueey-por-la-informatizacion-de-la-sociedad>. 2.

INTECO. 2009. *Ingeniería del software: Metodología y desarrollo de software*. España : s.n., 2009.

López, Eva Santana. 2003. *Las promociones publicitarias en la prensa. Evolución y análisis sistémico de las promociones realizadas entre los años 1990-1998 en la prensa de información general de Catalunya*. España, 2003.

Luis Calabria, Pablo Piriz. 2003. *Metodología XP*. Uruguay : s.n., 2003.

Markus Völter, Heidenheim, Germany. 2005. *Software Architecture*. 2005.

Martín, David Andrés. 2009. *Música Industrial y promoción. ¿Cómo ha cambiado el marketing musical?* 2009.

Martínez, Víctor Martínez. 2015. *LA PUBLICIDAD EN LA INDUSTRIA MUSICAL. LA COMUNICACIÓN Y LA PROMOCIÓN DE LA MÚSICA EN VIVO EN LAS*. 2015.

Metodologías de desarrollo de software. **Sanchez, María A. Mendoza. 2004.** Perú : s.n., 2004.

Microsoft Developer Network. 2012. Microsoft Developer Network. Revisiones de código y estándares de codificación. [En línea] 2012. <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.

Mora, Sergio Luján. 2002. *Programación de Aplicaciones Web: Historia, Principios básicos y Cliente Web*. Alicante : Club Univeristario, 2002.

Morales, Flores Elusa. 2004. Bibliodocencia. [En línea] 2004. [Citado el: 4 de 6 de 2016.] http://www.bibliodocencia.com/4/4_6.pdf.

Moreno, Nancy González y González, Fermín O Rodríguez. 2002. *La gestión de la información como base a la gestión del conocimiento y del aprendizaje en las organizaciones en las universidades*. Habana : s.n., 2002.

Pearson Education. 2005. *Sommerville. Ingeniería de Software Séptima Edición*. Madrid, España : s.n., 2005.

Penadés, Patricio Letelier. 2010. *Metodologías ágiles para el desarrollo de software: Extrema Programming*. Valencia : s.n., 2010.

Potencier, F y Zaninotto, F. 2009. Symphony 1.2, la guía definitiva. 3 de 10 de 2009.

Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico. 5ta edición*.

Project Team Of JQuery. 2010. JQuery Project. [En línea] 2010. [Citado el: 03 de 11 de 2015.] <http://jquery.com>.

Revista del Instituto Tecnológico de Informática. **Molineda, Ramón. 2005.** Valencia : Borland, 2005.

Ron Morrison, Flavio Oquendo. 2005. *Software Architecture*. Italia : Board, 2005.

Santesmases, Miguel Mestre. 1996. *Términos de marketing. Diccionario-Base de Datos*. Madrid : Ediciones Pirámide, 1996.

Glosario de términos

AJAX: Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

Aplicación Web: Una aplicación web es cualquier aplicación que es accedida vía web por una red como internet o una intranet.

Arquitectura: Conjunto de decisiones significativas acerca de la organización de un sistema de software, la selección de los elementos estructurales a partir de los cuales se componen el sistema. La misma se interesa no solo por la estructura y el comportamiento, sino también por las restricciones y compromisos de uso, funcionalidad, funcionamiento, flexibilidad al cambio, reutilización, comprensión, economía y tecnología, así como por aspectos estéticos.

Framework: Se conoce como marco de trabajo y constituye un conjunto de conceptos, metodologías y herramientas de administración y diseño para el desarrollo de forma estandarizada de una aplicación.

HTML: Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

IDE: Integrated Development Enviroment (Entorno de desarrollo integrado).

Módulo: Encapsula un conjunto de funciones que debe realizar el sistema, las cuales son agrupadas por tener características muy similares y se definen en la etapa de diseño.

MVC: Modelo Vista Controlador, patrón de arquitectura de software.

ORM: Object-Relational Mapping (Mapeo objeto-relacional). Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

TIC: Tecnologías de la Informática y las Comunicaciones.