

Tabla de Contenido

1. Introducción	1
1.1. Contexto	1
1.1.1. PRASEDEC	1
1.1.2. Museos virtuales en Internet	2
1.2. Framework	4
1.3. Objetivos	4
1.3.1. Principales	4
1.3.2. Específicos	5
1.4. Contenidos de la Memoria	5
2. Marco Teórico	6
2.1. Soluciones actuales	6
2.2. Metodologías de desarrollo	8
2.3. Arquitecturas	8
2.3.1. Arquitectura de tres capas	9
2.3.2. Modelo-Vista-Controlador	10
2.4. Framework	11
2.4.1. Black-box	11
2.4.2. White-box	11
2.4.3. Gray-box	12
2.5. Tecnologías utilizadas	12
2.6. Seguridad informática	13
2.6.1. Ataque de acceso a control	14
2.6.2. Inyección de SQL	15
2.7. Usabilidad de interfaces	16
3. Especificaciones del Problema	18
3.1. Problema a resolver	18
3.2. Relevancia	19
3.3. Requisitos	19
3.3.1. Requisitos de usuario	20
3.3.2. Requisitos de software	24
3.3.3. Matriz de trazado	30
3.4. Criterios de aceptación	31
4. Descripción de la Solución	32

4.1. Arquitectura lógica	32
4.2. Diseño base de datos	34
4.2.1. Modelos Django	34
4.2.2. Herencia	36
4.2.3. Recursos	36
4.2.4. Exhibits	37
4.2.5. Opiniones	38
4.2.6. Exhibitions	39
4.3. Diseño de procesos	39
4.3.1. Diagrama de flujo de usuarios	40
4.3.2. Procesamiento de requests	42
4.4. Diseño interfaces	44
4.4.1. Django templates y extensibilidad	45
4.4.2. Programación orientada a objetos	47
4.4.3. Utilización de diccionarios	50
4.4.4. Usabilidad en las interfaces	51
4.5. Seguridad en Django	55
4.5.1. Formularios y validadores	55
4.5.2. Decoradores	56
5. Validación de la solución	58
5.1. Extensión Framework	58
5.1.1. Nuevos exhibits: Vídeo y PDF	58
5.1.2. Visualización visitante	63
5.2. Deployment	68
5.2.1. Configuración inicial	68
5.2.2. Configuración base de datos	68
5.2.3. Dependencias	69
5.2.4. Ejecución	70
5.3. Ejemplo y guidelines	70
5.4. Feedback humano y usabilidad	72
6. Conclusión	73
6.1. Resultados obtenidos y objetivos cumplidos	73
6.2. Análisis crítico de los resultados	73
6.3. Trabajo futuro	74
Bibliografía	75

Índice de Tablas

2.1. Tabla Comparativa entre Three.js, WhiteStormJS y BabylonJS.	7
2.2. Tabla comparativa entre frameworks basales.	13
3.1. RU001: Extensibilidad	20
3.2. RU002: Funcionalidad basal	21
3.3. RU003: Compatibilidad	21
3.4. RU004: Tipos contenido	21
3.5. RU005: Grupos usuarios	21
3.6. RU006: Datos usuarios	21
3.7. RU007: Información exhibiciones	21
3.8. RU008: Sencillez	22
3.9. RU009: Visualizar exhibiciones	22
3.10. RU010: Retroalimentación visitantes	22
3.11. RU011: Manejar opiniones	22
3.12. RU012: Administrar recursos	22
3.13. RU013: Facilidad de ver recursos	22
3.14. RU014: Administrar exhibits	23
3.15. RU015: Información de uso de exhibits	23
3.16. RU016: Manejo agenda	23
3.17. RU017: Modificación agenda	23
3.18. RU018: Accesibilidad exhibiciones	23
3.19. RU019: Facilidad de ejecución	23
3.20. RU020: Independencia de sistema operativo	24
3.21. RS001: Buenas prácticas	24
3.22. RS002: Documentación	24
3.23. RS003: Vistas con datos reales	24
3.24. RS004: Vistas optimizadas para computadores de escritorio	24
3.25. RS005: Independencia de navegador	25
3.26. RS006: Soporte múltiples exhibits	25
3.27. RS007: Extensibilidad exhibits	25
3.28. RS008: Sistema de usuarios	25
3.29. RS009: Grupos de usuarios	25
3.30. RS010: Modificación datos personales	26
3.31. RS011: Visualización de agenda	26
3.32. RS012: Usabilidad en vista visitante	26
3.33. RS013: Aplicación maneja las visualizaciones	26

3.34. RS014: Sistema de opiniones	26
3.35. RS015: Validación de opiniones	27
3.36. RS016: Manejo de opiniones	27
3.37. RS017: Manejo y extensibilidad recursos	27
3.38. RS018: Vista administración recursos	27
3.39. RS019: Validación recursos y exhibits	28
3.40. RS020: Filtrado recursos	28
3.41. RS021: Vista administración exhibits	28
3.42. RS022: Información de uso de exhibits	28
3.43. RS023: Administración agenda	28
3.44. RS024: Múltiples exhibit por exhibición	29
3.45. RS025: Edición total de exhibiciones	29
3.46. RS026: Visibilidad exhibiciones	29
3.47. RS027: Scripts de ejecución del framework	29
3.48. RS028: Independencia de sistema operativo	29

Índice de Ilustraciones

1.1. Museo virtual a base de fotografías panorámicas.	3
2.1. Arquitectura de tres capas.	9
2.2. Arquitectura de tres capas escalada horizontalmente.	10
3.1. Matriz de trazado de requisitos de usuario vs requisitos de software.	30
4.1. Estructura física del proyecto.	33
4.2. Ejemplo entidad base de datos.	35
4.3. Ejemplo de herencia doble en la base de datos.	36
4.4. Diagrama relacional recursos.	37
4.5. Diagrama relacional exhibits.	38
4.6. Tabla de opiniones.	39
4.7. Jerarquía de exhibits.	39
4.8. Diagrama relacional Exhibitions.	40
4.9. Primer boceto de diagrama de flujo del Framework.	41
4.10. Diagrama de flujo de vistas comunes.	42
4.11. Diagrama de flujo de vistas curador.	43
4.12. Diagrama de secuencia de un request.	44
4.13. Vista de recursos de audio.	46
4.14. Vista de recursos de modelos.	47
4.15. Vista de recursos de imágenes.	48
4.16. Templates reutilizados utilizando { % include % } para ver recursos.. . . .	48
4.17. Contraste de Memorability entre vistas.	52
4.18. Bloques en una vista.	52
4.19. Vistas con acciones específicas por elemento.	53
4.20. Barras de navegación de usuarios con distintos permisos.	53
4.21. Vistas jerárquicas para ver exhibits.	54
4.22. Vista de visualización de exhibit.	54
5.1. Formulario de subida de exhibit tipo Unity.	59
5.2. Menú de navegación incluyendo nuevos exhibits.	60
5.3. Formularios de subida de exhibits tipo Video y PDF.	61
5.4. Previsualización de un PDFExhibit desde la vista de curadores.	63
5.5. Página inicial Framework.	64
5.6. Página inicial Framework extendido.	65
5.7. Nueva barra de navegación visitante.	66

5.8. Nueva vista de exhibiciones.	67
5.9. Formulario de opinión de visitantes.	67
5.10. Opciones de compilación de Unity.	72

Índice de Códigos

2.1. Ejemplo JSON usuarios.	14
2.2. Ejemplo consulta SQL.	15
2.3. Ejemplo inyección consulta SQL.	15
4.1. Ejemplo de una clase vista.	34
4.2. Ejemplo implementación de Django Model.	35
4.3. Ejemplo consultas Django.	35
4.4. Ejemplo URL Dispatcher.	43
4.5. URL página contacto.	43
4.6. Template base del proyecto.	46
4.7. Extensión de template.	47
4.8. Implementación de herencia en clases vista.	49
4.9. Anidación de diccionarios y funciones como variables.	50
4.10. Diccionario que almacena la información de cada recurso.	51
4.11. Clase vista con métodos GET y POST.	55
4.12. Clase que representa un formulario.	56
4.13. Validación de un formulario para la clase <code>ExternalTemplate</code>	56
4.14. Ejemplos de decoradores.	57
4.15. Decorador de comprobación de grupos.	57
5.1. Clases asociadas al exhibit Unity.	59
5.2. Clases VideoExhibit y PDFExhibit.	60
5.3. Clases asociadas al formulario de UnityExhibit.	61
5.4. Clases asociadas al formulario de PDFExhibit.	61
5.5. Nuevas clases de vista para los formularios de video y pdf.	62
5.6. Asociación de vistas de añadir video y pdf en el URL Dispatcher.	62
5.7. Diccionario <code>MUSEUMS_TYPES</code> y los métodos para el tipo pdf.	62
5.8. Utilización de ID para agregar el vínculo a <i>enviar opinión</i>	66
5.9. Configuración base de datos en Django	69
5.10. Configuración <code>virtualenv</code>	69