



UNIVERSIDAD TÉCNICA DEL NORTE

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**“ESTUDIO DEL FRAMEWORK YII 2 PARA LA IMPLEMENTACIÓN DE UNA
PLATAFORMA WEB ORIENTADA A SOLUCIONES EMPRESARIALES.
DESARROLLO DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL
FASE 2”**

AUTOR: MAURICIO ROBINSON CHAMORRO CHAMORRO

DIRECTORA: ING. CATHY GUEVARA

IBARRA – ECUADOR

2016



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004042469		
APELLIDOS Y NOMBRES:	CHAMORRO CHAMORRO MAURICIO ROBINSON		
DIRECCIÓN:	VICTOR JARAMILLO 1111 Y RAÚL MONTALVO		
EMAIL:	unrealmach@hotmail.com		
TELÉFONO FIJO:	062558296	TELÉFONO MÓVIL:	0996683742

DATOS DE LA OBRA	
TÍTULO:	ESTUDIO DEL FRAMEWORK YII 2 PARA LA IMPLEMENTACIÓN DE UNA PLATAFORMA WEB ORIENTADA A SOLUCIONES EMPRESARIALES. DESARROLLO DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE 2
AUTOR (ES):	CHAMORRO CHAMORRO MAURICIO ROBINSON
FECHA: AAAAMMDD	
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR /DIRECTOR:	MGS. CATHY GUEVARA

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, MAURICIO ROBINSON CHAMORRO CHAMORRO, con cédula de identidad Nro. 1004042469, en calidad de autor (es) y titular (es) de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los días del mes de ...Julio..... de 20.16.....

EL AUTOR:

(Firma).....

Nombre: MAURICIO ROBINSON CHAMORRO CHAMORRO



UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, MAURICIO ROBINSON CHAMORRO CHAMORRO, con cédula de identidad Nro. 1004042469, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado: ESTUDIO DEL FRAMEWORK YII 2 PARA LA IMPLEMENTACIÓN DE UNA PLATAFORMA WEB ORIENTADA A SOLUCIONES EMPRESARIALES. DESARROLLO DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE 2, que ha sido desarrollado para optar por el título de: INGENIERO EN SISTEMAS COMPUTACIONALES en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, a los días del mes de ..Julio..... de 20..16..

(Firma) 

Nombre: MAURICIO ROBINSON CHAMORRO CHAMORRO

Cédula: 1004042469



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

Certifico que la Tesis previa a la obtención del título de Ingeniería en Sistemas Computacionales con el tema “ESTUDIO DEL FRAMEWORK YII 2 PARA LA IMPLEMENTACIÓN DE UNA PLATAFORMA WEB ORIENTADA A SOLUCIONES EMPRESARIALES. DESARROLLO DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE 2” ha sido desarrollada y terminada en su totalidad por el Sr. Mauricio Robinson Chamorro Chamorro con C.I. 1004042469 bajo mi supervisión para lo cual firmo en constancia

Atentamente,

Ing. Cathy Guevara
DIRECTORA DE TESIS



Gobierno Autónomo
Descentralizado Municipal de
San Miguel de Ibarra



IBARRA
cerca de nuestra gente

LA SUSCRITA DRA. ROCÍO OLIVO

CERTIFICA:

Que: El señor **MAURICIO ROBINSON CHAMORRO CHAMORRO**, con cédula de ciudadanía 1004042469, desarrollo su trabajo de grado con el tema: "ESTUDIO DEL FRAMEWORK YII 2 PARA LA IMPLEMENTACION DE UNA PLATAFORMA WEB ORIENTADA A SOLUCIONES EMPRESARIALES. DESARROLLO DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE 2", por lo que nos es grato informar que cumplió con satisfacción las pruebas funcionales y de requerimientos, por lo que se recibió el proyecto como culminado y realizado por parte del señor arriba mencionado

Es cuanto puedo certificar, facultando al interesado hacer uso del presente documento, para fines pertinentes en la Universidad Técnica del Norte.

Ibarra, 17 de mayo del 2016

Atentamente,

Dra. Rocío Olivo

DIRECCIÓN DE INCLUSIÓN SOCIAL

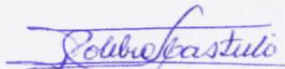


Ibarra, 04 de mayo de 2016

CERTIFICACIÓN

Certifico que la Tesis previa a la obtención del título de Ingeniería en Sistemas Computacionales con el tema "ESTUDIO DEL FRAMEWORK YII 2 PARA LA IMPLEMENTACIÓN DE UNA PLATAFORMA WEB ORIENTADA A SOLUCIONES EMPRESARIALES. DESARROLLO DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE 2" referente a los módulos Bitácora de Consumo y Generación de Recetas ha sido desarrollada y terminada en su totalidad por el Sr. Mauricio Robinson Chamorro Chamorro con C.I. 1004042469 bajo mi supervisión para lo cual firmo en constancia

Atentamente,



Dra. Susana Castillo Lara
TÉCNICO ÁREA DE NUTRICIÓN



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

Dedicatoria

Este trabajo está dedicado a mis padres Raúl Chamorro y Claudina Chamorro, quienes me apoyaron incansablemente para ver realizado el objetivo de culminar mis estudios.

A mis hermanos por su apoyo a lo largo de todo este tiempo que duró mis estudios.

A la Ing. Sofía Mejía quien gracias a su apoyo, decisión y persistencia se pudo culminar el software Sinuti

Mauricio Chamorro.



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

Agradecimiento

Agradezco a Dios por la vida, por todas las bendiciones que me ha otorgado y por su amor infinito.

Agradezco a la Universidad Técnica del Norte que me abrió sus puertas para realizar mis estudios superiores.

Agradezco a mi directora de Tesis la Ing. Cathy Guevara por su valiosa ayuda en la realización de este trabajo.

Agradezco a todos aquellos que fueron mis profesores por haber compartido sus conocimientos conmigo.

Agradezco al Lic. Felipe Bixby, a la Dra. Romelia Goyes y a la Dra. Susana Castillo Lara por aportar con sus conocimientos y poder así culminar con estándares de calidad el presente trabajo.

Agradezco a la Dra. Rocío Olivo quien abrió las puertas de su institución para poder desarrollar el proyecto.

Mauricio Chamorro.

Resumen

El campo del desarrollo de aplicaciones web está abriendo posibilidades para la construcción de nuevas herramientas que prometan solucionar los requerimientos del mundo actual.

El propósito de este proyecto de grado es realizar un estudio de uno de los Frameworks PHP que ha surgido los últimos años, con el objetivo de demostrar que Yii 2 es un Framework apto para el desarrollo de aplicaciones empresariales aplicado al mejoramiento de las características que brinda el Sistema nutrimental infantil (Sinuti) y a su vez la creación de documentación sobre dicho Framework que ayudará a los interesados en aprender esta herramienta en menor tiempo.

En el capítulo uno; se realiza una introducción general del proyecto de grado en el que se detalla el problema, los objetivos, el alcance y la justificación proyecto.

En el capítulo dos; se realiza el marco teórico del Framework y del aplicativo a desarrollar, se detalla el ámbito social sobre el cual se desarrolla la fase 2 de Sinuti, la arquitectura, Templates, ciclo de vida y características que componen al Framework.

En el capítulo tres, se explica la Fase 2 de Sinuti con la ayuda de la metodología XP y se da a conocer información acerca del manejo de widgets, seguridad, configuración para el entorno de desarrollo y web services tipo RestFul.

En el capítulo cuatro y cinco, se expone la implementación del sistema según la metodología XP.

En el capítulo seis, se muestra un análisis comparativo del Framework Yii 2 vs Yii 1; así como el análisis de impactos, conclusiones y recomendaciones del proyecto.

Summary

The area of web application development is opening possibilities for building new tools focused to solve the requirements of today's world.

The purpose of this degree project was a study on one of the PHP frameworks that has been developed in recent years like Yii 2, which aims to demonstrate its effectiveness in the development of business software, this was applied to the improvement of the features provided by the “Sistema Nutricional Infantil” (Sinuti) and at the same time created documentation about the framework that would allow stakeholders to learn this tool in minor time.

In the chapter one a general introduction of the project was done. There were detailed the problem, goals, scope and justification of the research.

In the chapter two the theoretical framework and the developed application were done. There were detailed the social scope on which the phase 2 of Sinuti is based, the architecture, templates, life cycle and characteristics that made-up the framework.

In chapter three, phase 2 of Sinuti was explained by means of the XP methodology and information about widgets, security management, configuration for the environmental development and RESTful web services were provided.

In chapter four and five the implementation of the system according to the XP methodology was explained.

Finally, the chapter six presents a comparative analysis of the Yii framework 2 vs Yii 1, as well as the impact analysis, conclusions and recommendations of the project.

TABLA DE CONTENIDO

CAPÍTULO I	1
1. INTRODUCCIÓN	1
1.1. PROBLEMA	1
1.1.1. ANTECEDENTES	1
1.1.2. SITUACIÓN ACTUAL	1
1.1.3. PROSPECTIVA	2
1.1.4. PLANTEAMIENTO DEL PROBLEMA	2
1.2. OBJETIVOS	3
1.2.1. OBJETIVO GENERAL	3
1.2.2. OBJETIVOS ESPECÍFICOS	3
1.3. ALCANCE Y LIMITACIONES	4
1.3.1. ALCANCE	4
1.3.2. LIMITACION	4
1.3.3. HERRAMIENTAS DE DESARROLLO	4
1.3.4. DIAGRAMA DE RED	5
1.3.5. DIAGRAMA DE LOS MÓDULOS DE LA APLICACIÓN	5
1.4. JUSTIFICACIÓN	6
2. MARCO TEÓRICO	8
2.1. ESTADO DEL SEGUIMIENTO NUTRIMENTAL INFANTIL EN EL ECUADOR	8
2.1.1. DEFINICIÓN DEL SEGUIMIENTO NUTRIMENTAL INFANTIL EN LOS CIBV	8
2.1.2. PROCEDIMIENTOS DE LAS ENTIDADES DE SALUD	9
2.2. SITUACIÓN ACTUAL DE APLICACIONES EN YII 2	9
2.2.1. CARACTERÍSTICAS DE YII 2	11
2.2.1.1. BASIC TEMPLATE	12
2.2.1.2. ADVANCED TEMPLATE	12
2.2.2. CICLO DE VIDA DE LAS PETICIONES	13
2.2.3. VENTAJAS DE DESARROLLAR CON YII 2	15
2.3. PATRÓN DE DISEÑO MVC	16
2.3.1. MODELO	16
2.3.2. VISTA	16
2.3.3. CONTROLADOR	17
2.4. MVC EN YII 2	17
2.4.1. VISTA EN YII 2	19
2.4.1.1. ESTRUCTURA DE LAS VISTAS	19
2.4.1.2. THEMES	22
2.4.2. MODELO EN YII 2	23
2.4.2.1. OBJETOS DE ACCESO A LA BASE DE DATOS (DAO)	23
2.4.2.2. QUERY BUILDER	24
2.4.2.3. ACTIVE RECORD ORM	25
2.4.2.4. ESTRUCTURA DEL MODELO	27

2.4.3.	CONTROLADORES Y ENRUTAMIENTO EN YII 2	28
2.4.3.1.	ACCIONES	28
2.4.4.	VALIDACIONES DEL MVC EN YII 2	29
2.5.	COMPONENTES DE APLICACIÓN DE YII 2	30
2.5.1.	COMPONENTES	30
2.5.2.	GENERADOR DE CÓDIGO	31
2.5.3.	CONTROL DE ACCESO BASADO EN ROLES (RBAC)	32
2.5.3.1.	RBAC EN YII 2	32
2.6.	CARACTERÍSTICAS WEB RESPONSIVE	32
CAPÍTULO III		34
3.	DESARROLLO DE LA FASE DOS DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL Y DOCUMENTACIÓN TÉCNICA DE YII 2	34
3.1.	SITUACIÓN ACTUAL DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE UNO	34
3.2.	ACUERDO MIES Y GAD	34
3.3.	METODOLOGÍA XP	34
3.3.1.	PLANEACIÓN	35
3.3.1.1.	ANÁLISIS DE REQUERIMIENTOS	35
3.3.2.	DISEÑO	36
3.3.3.	CODIFICACIÓN	37
3.3.4.	PRUEBA	37
3.4.	DOCUMENTACIÓN TÉCNICA DE YII 2	37
3.4.1.	DESPLIEGUE DE YII 2 EN ENTORNO DE DESARROLLO Y PRODUCCIÓN	37
3.4.1.1.	CONFIGURACIÓN DE ENTORNOS FRONTEND Y BACKEND EN LA FASE DE DESARROLLO	39
3.4.2.	MANEJO DE WIDGETS DEL CORE DE YII 2	44
3.4.3.	USO DE CACHE	49
3.4.4.	SEGURIDAD	52
3.4.4.1.	FILTROS DE CONTROL DE ACCESO (ACF)	52
3.4.4.2.	USO DEL CONTROL DE ACCESO BASADO EN ROLES (RBAC)	53
3.4.4.3.	CONFIGURACIÓN DEL MÓDULO DE ADMINISTRACIÓN DE RBAC	56
3.4.5.	RESTFUL WEB SERVICES EN YII 2	58
3.4.5.1.	CONFIGURACIÓN DEL ENTORNO PARA API YII 2	59
3.4.5.2.	USO DE YII2 PARA CLIENTE REST	63
3.4.6.	HERRAMIENTAS DE DESARROLLO	64
CAPÍTULO IV		65
4.	IMPLEMENTACIÓN DE SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE DOS	65
4.1.	FASE DE PLANIFICACIÓN	65
4.1.1.	HISTORIAS DE USUARIO	65
4.1.1.1.	HISTORIAS DE USUARIO GASTRÓNOMO	65
4.1.1.2.	HISTORIA DE USUARIO RBAC	67
4.1.1.3.	HISTORIAS DE USUARIO NUTRICIONISTA	68

4.1.2.	DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE	69
4.2.	FASE DE DISEÑO.....	70
4.2.1.	CASOS DE USO.....	70
4.2.2.	DIAGRAMAS DE PROCESO.....	73
4.2.2.1.	DIAGRAMA DE PROCESO GESTIÓN DE RECETAS.....	74
4.2.2.2.	DIAGRAMA DE PROCESO GESTION DE TIPO DE RECETA.....	75
4.2.2.3.	DIAGRAMA DE PROCESOS GESTION DE PERMISOS PARA ROLES.....	76
4.2.2.4.	DIAGRAMA DE PROCESOS ASIGNACION DE CARTA SEMANAL A INFANTES.....	77
4.2.3.	DIAGRAMA ENTIDAD RELACION.....	79
	CAPÍTULO V.....	80
5.	FASE DE CODIFICACIÓN.....	80
5.1.1.	PLAN DE ENTREGAS.....	80
5.2.	FASE DE PRUEBAS.....	80
5.2.1.	PRUEBAS GESTION MÓDULO GASTRONOMÍA.....	80
5.2.2.	PRUEBAS GESTIÓN MÓDULO RBAC.....	84
5.2.3.	PRUEBAS GESTION MÓDULO BITÁCORA.....	86
	CAPÍTULO VI.....	88
6.	ANÁLISIS DE IMPACTOS.....	88
6.1.	ANÁLISIS COMPARATIVO DEL FRAMEWORK YII 2 VS YII 1.1.16.....	88
6.2.	ANÁLISIS DE IMPACTOS EN EL SERVICIO DE LOS CIBV DEL SISTEMA NUTRIMENTAL INFANTIL FASE DOS.....	89
6.3.	CONCLUSIONES.....	91
6.4.	RECOMENDACIONES.....	92
	BIBLIOGRAFÍA.....	93
	ANEXOS.....	94

INDICE DE TABLAS

Tabla 1: Descripción de la arquitectura de Yii 2 con MVC.....	18
Tabla 2: Formas de acceso (Lazy y Eager Loading).....	27
Tabla 3: Métodos de renderización en un controlador	29
Tabla 4: Componentes del framework Yii 2	31
Tabla 5: Tipos de cache que soporta Yii 2.....	49
Tabla 6: Api para uso del cache en Yii 2	51
Tabla 7: Historia de usuario para la gestión de recetas	65
Tabla 8: Historia de usuario que indica la categorización de porciones.....	66
Tabla 9: Historia de usuario para el Rbac	67
Tabla 10: Historia de usuario del módulo bitácora de consumo	68
Tabla 11: Historia de usuario que indica el procedimiento del porcentaje de adecuación	69
Tabla 12: Descripción caso de uso gestión módulo gastronomía.....	70
Tabla 13: Descripción caso de uso gestión Rbac	72
Tabla 14: Descripción caso de uso gestión bitácora.....	73
Tabla 15: Caso de prueba crear nuevo receta.....	80
Tabla 16: Caso de prueba actualizar receta	81
Tabla 17: Caso de prueba eliminar receta	82
Tabla 18: Caso de prueba ingreso de tipo de receta	82
Tabla 19: Caso de prueba actualizar tipo receta	83
Tabla 20: Caso de prueba eliminar tipo receta	84
Tabla 21: Caso de prueba crear nuevo permiso y actualizarlo	84
Tabla 22: Caso de prueba eliminar permiso	85
Tabla 23: Caso de prueba crear nuevo rol y actualizarlo	85
Tabla 24: Caso de prueba eliminar permiso	86
Tabla 25: Caso de prueba asignar carta semanal a infantes	86
Tabla 26: Resultados benchmarking frameworks PHP	89

INDICE DE FIGURAS

Figura 1: Diagrama de red	5
Figura 2: Fases y módulos a desarrollar del sistema de Seguimiento Nutricional Infantil	5
Figura 3: Búsqueda de frameworks PHP según Google Trends revisado el 29-01-2016	10
Figura 4: Porcentaje de información en Internet de Yii 2 según Google Search.....	10
Figura 5: Flujo de la gestión de una aplicación	13
Figura 6: Esquema de MVC.....	16
Figura 7: Esquema MVC Yii 2	17
Figura 8: Esquema de vistas Yii 2	19
Figura 9: Esquema del layout del theme AdmLte	20
Figura 10: Theme idea_v1.3 integrado al framework.....	22
Figura 11: Uso de Query Builder y declaración generada por el mismo	24
Figura 12: Comparación entre Active Record y DAO para la inserción de un nuevo registro en la tabla Customer. Superior: Uso de Active Record. Inferior: Uso de DAO.	25
Figura 13: Modelo relacional one-to-many de ejemplo	26
Figura 14: Relaciones que se crean en los modelos de ejemplo	26
Figura 15: Uso de relaciones anidadas	26
Figura 16: Objetos para gestión del modelo	28
Figura 17: Ciclo de vida de XP	35
Figura 18: Selección de entornos en Yii 2 para una aplicación con Template avanzado.....	37
Figura 19: Estructura de entornos en Yii 2 con Template avanzado	38
Figura 20: Activación de host virtual	39
Figura 21: Configuración común para un host virtual	40
Figura 22: Configuración en el fichero hosts de Windows	40
Figura 23: Ficheros de configuración del servidor en un Template avanzado Yii 2	41
Figura 24: Fichero .htaccess que permite la redirección del Backend y Frontend en un mismo dominio	41
Figura 25: Fichero .htaccess que permite la eliminación de index.php	42
Figura 26: Fichero main.php del Backend que indica las nuevas reglas de redirección.....	42
Figura 27: Fichero main.php del Frontend que indica las nuevas reglas de redirección	43
Figura 28: Backend y Frontend en un mismo dominio con un Template avanzado de Yii 2	43
Figura 29: Ubicación de los widgets del core de Yii 2	44
Figura 30: Configuración avanzada del widget ActiveForm de Yii 2.....	45
Figura 31: Widget ActiveForm de Yii 2	45
Figura 32: Envío de alertas desde el controlador hacia la vista a través del objeto de sesión	46
Figura 33: Configuración del widget Alert	46
Figura 34: Función Javascript para llamar al widget de modal	47
Figura 35: Ejemplo del widget de navegación	47
Figura 36: Configuración del widget Nav	48
Figura 37: Ejemplo del widget NavBar	48
Figura 38: Configuración del widget NavBar.....	49
Figura 39: Uso del componente de cache para consultas de bases de datos.....	52
Figura 40: Configuraciones para el uso del ACF	52
Figura 41: Diagrama de ejemplo de la permisología del RBAC en Yii 2	54
Figura 42: Método para validar los permisos del Rbac	55
Figura 43: Configuración del método can() en las acciones del controlador	55
Figura 44: Configuración del componente authManager	56
Figura 45: Configuración del módulo para administración del Rbac	57
Figura 46: Vista del módulo de administración del RBAC Yii 2-admin	57
Figura 47: Esquema de aplicaciones que usan web services	58
Figura 48: Esquema completo de comunicación entre un Frontend y un Api para Yii 2.....	59
Figura 49: Base de datos de ejemplo para Api Yii 2	59
Figura 50: Configuración del entorno para API paso 1	60

Figura 51: Configuración del entorno para API paso 2.....	60
Figura 52: Estructura del módulo v1 para el entorno de API.....	61
Figura 53: Lista de ejemplo con las operaciones del API Rest de Yii 2.....	61
Figura 54: Configuración de interfaz web service tipo Rest en Yii 2.....	62
Figura 55: Configuración de config.php en el entorno Api para disponer de los servicios web.....	62
Figura 56: Configuración de un action para el consumo de un web service tipo Rest.....	63
Figura 57: Caso de uso gestión módulo gastronomía.....	70
Figura 58: Gestión módulo Rbac.....	71
Figura 59: Gestión módulo Bitácora.....	73
Figura 60: Diagrama de proceso gestión recetas.....	74
Figura 61: Diagrama de proceso gestión tipo receta.....	75
Figura 62: Diagrama de proceso gestión de los permisos para roles.....	76
Figura 63: Diagrama de procesos para la asignación de carta semanal a infantes.....	77
Figura 64: Diagrama de procesos registro de bitácora.....	78
Figura 65: Diagrama entidad relación Sinuti.....	79
Figura 66: Plan de entregas Sinuti fase 2.....	80
Figura 67: Resultados de eficiencia en procesamiento de peticiones.....	88
Figura 68: Resultado de la eficiencia del ORM.....	89

CAPÍTULO I

1. INTRODUCCIÓN

1.1. PROBLEMA

1.1.1. ANTECEDENTES

Con la aparición del internet y el constante crecimiento de la accesibilidad a esta tecnología, las empresas y organizaciones tienen la posibilidad de ofrecer sus productos y servicios a nivel mundial.

Una aplicación web permite tener una implementación y desarrollo centralizado. Con el uso de internet las organizaciones y personas pueden mantener la conectividad y trabajar en una red privada sin invertir en costos de infraestructura de telecomunicaciones (Ramírez, 2012).

Desde la ciencia de la informática se ha generado una constante mejora en cuanto a librerías, patrones de diseño y estándares para la codificación, lo que ha mejorado la eficiencia, estructura, velocidad y seguridad de las aplicaciones siendo estas las prestaciones de los marcos de trabajo o frameworks (Acens, 2016).

En la actualidad muchas empresas dedicadas al desarrollo de aplicaciones y servicios en la web ven en los frameworks una herramienta indispensable para generar ganancias en menos tiempo. Sin embargo, cada framework varía en la cantidad de tiempo para llegar a manejarlo con eficiencia ya sea por el lenguaje, complejidad, tecnologías o poca documentación en el idioma nativo del interesado.

1.1.2. SITUACIÓN ACTUAL

Yii 2 es un framework que se lanzó a la web el 12 de octubre del 2014, posee la ventaja de incorporar las últimas tecnologías en lenguajes de programación para la web como: Css3¹, Html5², PHP³ 5.4 y compatibilidad con PHP 7, una herramienta para la gestión de dependencias de PHP (Composer⁴), fácil configuración de web services RESTful API⁵, trabaja con distintas bases de datos relacionales y noSQL⁶ y está habilitado para pruebas unitarias. Gracias a su arquitectura basada en componentes, Yii 2 se emplea para el desarrollo

de aplicaciones a gran escala como: portales, foros, sistemas de gestión de contenidos (CMS⁷), proyectos de e-commerce⁸, sistemas de gestión de relaciones con los clientes (CRM⁹) y planificación de recursos empresariales (ERP¹⁰).

1.1.3. PROSPECTIVA

Con el estudio a realizarse se generará la documentación necesaria para que estudiantes y profesionales interesados en el estudio de Yii 2 puedan incrementar sus conocimientos sobre esta herramienta.

Se agregarán 3 módulos: control de acceso basado en roles (Rbac¹¹), bitácora de consumo y gastronomía al sistema de seguimiento nutricional infantil fase uno realizado por la Srta. Amanda Sofía Mejía Andrade estudiante de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte (UTN). Estos módulos mejorarán las características y prestaciones en cuanto a seguridad, roles de usuarios, preparación de recetas y un mejor control y seguimiento de las dietas alimenticias que los infantes ingieren en los Centros infantiles del buen vivir.

1.1.4. PLANTEAMIENTO DEL PROBLEMA

Uno de los principales servicios de los centros de desarrollo integral infantil es brindar una alimentación nutritiva a los infantes que asisten a los mismos por medio de una planificación semanal de alimentación, lo cual permite el desarrollo de destrezas y habilidades que ayudan al aprendizaje, crecimiento y reducen el riesgo de contraer enfermedades. Sin embargo en muchos de los casos debido a costumbres alimenticias y patologías los infantes no consumen en su totalidad la planificación de alimentación, lo que perjudica su correcto crecimiento y desarrollo. Además el Sistema Nutricional Infantil (Sinuti) no posee las características necesarias para el manejo óptimo de roles de usuario ni la capacidad de agregar nuevas recetas o preparaciones de la localidad para su respectivo control nutricional.

Debido a esto, se pretende desarrollar con el framework Yii 2 la fase 2 de Sinuti que permitirá a los centros infantiles: generar una bitácora de consumo para cada infante, agregar observaciones de reacciones alimentarias y el porcentaje de la alimentación que el infante consumió. Con esta información las autoridades podrán tomar las medidas necesarias para mejorar la nutrición de los infantes. También se agregará el control de acceso basado en roles (RBAC) para poder controlar y supervisar las diferentes actividades que los usuarios realicen

en la plataforma, así como la preparación de recetas con el fin de poder agregar platos típicos de la localidad a los procedimientos nutrimentales que realiza el sistema. Por otra parte, en el presente estudio se creará la documentación en español del framework Yii 2 como un aporte para que novatos y profesionales en el desarrollo de aplicaciones web de habla hispana, puedan disminuir el tiempo de desarrollo en este tipo de framework.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Estudiar el framework Yii 2 para la implementación de una plataforma web orientada a soluciones empresariales. Desarrollo del sistema de seguimiento nutrimental infantil fase 2.

1.2.2. OBJETIVOS ESPECÍFICOS

- Diagnosticar la situación actual de Yii 2.
- Implementar un sistema que mejorará la prestación de los servicios de preparación de recetas, roles de usuario y control alimenticio desarrollado en Yii 2 para el sistema de seguimiento nutrimental infantil usando metodología XP.
- Construir la documentación técnica que servirá de guía para la población interesada en el framework.
- Realizar un análisis de impactos del uso de la herramienta con respecto a anteriores versiones.

1.3. ALCANCE Y LIMITACIONES

1.3.1. ALCANCE

El presente estudio proporcionará a los programadores interesados en la creación de aplicaciones web con el framework Yii 2 una guía que facilite el aprendizaje y desarrollo del mismo. Además, permitirá a los centros infantiles usar el control de acceso basado en roles, las preparaciones de recetas, las porciones alimenticias y el control del consumo diario de alimentación de los infantes a través de una bitácora.

1.3.2. LIMITACION

Para la presente investigación se necesita de ebooks que se comprarán vía internet, ya que no se encuentra bibliografía en físico en las librerías o bibliotecas del país.

Es necesario tener conocimientos de administración de servidores web con Apache para poder configurar el entorno de producción con el framework. Así como tener conocimiento con el lenguaje de programación PHP para el servidor y HTML5, Css3 y JQuery para el cliente.

1.3.3. HERRAMIENTAS DE DESARROLLO

YII 2 tiene integrado en su ambiente de desarrollo los lenguajes de programación tales como: Css3, HTML5 y JQuery¹². La infraestructura cliente servidor será provista por el conjunto de herramientas del entorno de desarrollo WAMP¹³ Server. La implementación se realizará sobre Windows 7 con el servidor web Apache integrado al gestor de base de datos MySQL sobre lenguaje PHP. En la fase de testeo y producción se utilizará el sistema operativo Centos¹⁴ 6.5 con el entorno de desarrollo LAMP¹⁵.

1.3.4. DIAGRAMA DE RED

A continuación, se presenta un esquema del diagrama de red de la aplicación en la cual se implementará la fase dos del sistema de seguimiento nutricional infantil.

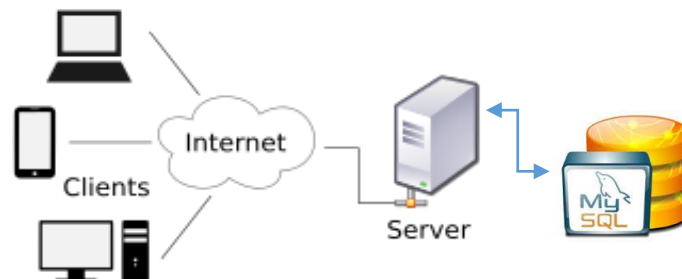


Figura 1: Diagrama de red

Fuente: Propio

1.3.5. DIAGRAMA DE LOS MÓDULOS DE LA APLICACIÓN

La Figura distingue las fases a desarrollar en el Sistema de seguimiento nutricional infantil.

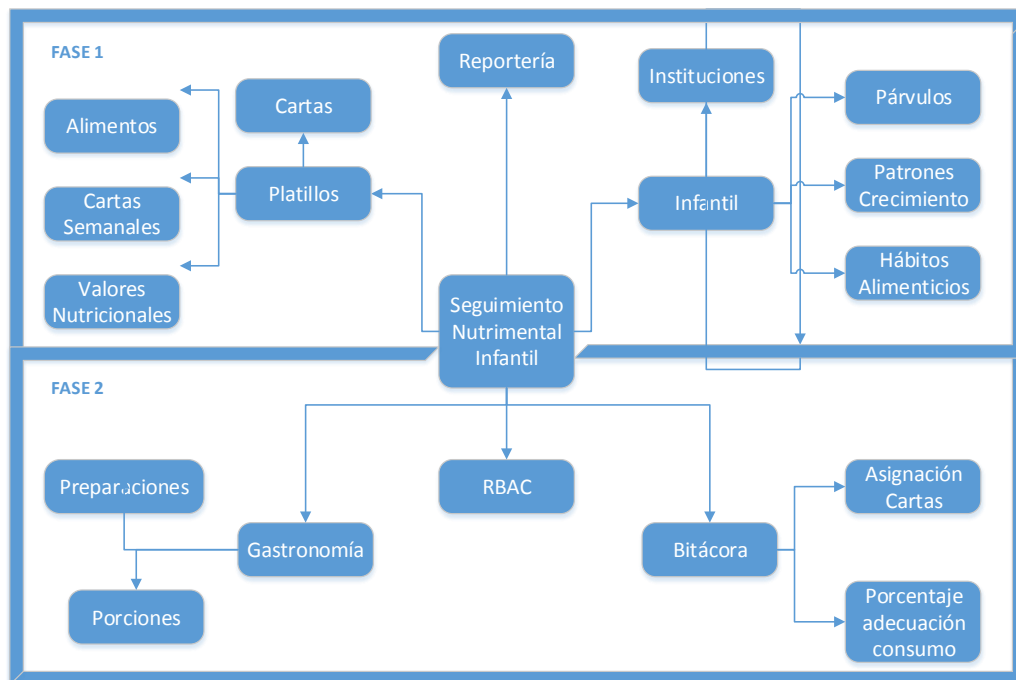


Figura 2: Fases y módulos a desarrollar del sistema nutricional infantil

Fuente: Propio

Este trabajo se enfoca en desarrollar los módulos correspondientes a la segunda fase del sistema nutricional infantil con el framework Yii 2, los cuales se describen a continuación:

Módulo de Gastronomía

Se compone de:

- **Preparaciones:** gestionará los platillos o recetas que posteriormente formaran parte de menús de la localidad.
- **Porciones:** gestionará las porciones para los alimentos y preparaciones, esto permitirá tener medidas acordes de las cantidades de alimentos para la población infantil.

Módulo Rbac (Control de acceso basado en roles)

Permitirá gestionar de manera correcta los distintos usuarios del sistema asignándoles acciones específicas de acuerdo a su rol.

Módulo de Bitácora

Se compone de:

- **Asignación de cartas:** gestionará la asignación de cartas o menús a los infantes, de acuerdo a políticas de los Cibv¹⁶.
- **Porcentaje adecuación consumo:** gestionará los indicadores de consumo alimenticio que ayudan a identificar problemas en las dietas alimentarias de los infantes por cada semana.

1.4. JUSTIFICACIÓN

De acuerdo al análisis de tendencias a través de la herramienta de comparación de popularidad de búsquedas de Google¹⁷ (Google Trends) realizada con términos referentes a frameworks PHP, se observa un incremento constante de la tendencia de popularidad de Yii 2, así como una disminución progresiva de las búsquedas referentes a los otros frameworks. Esto quiere decir que la población interesada en desarrollo con frameworks PHP está siempre en búsqueda de las nuevas herramientas innovadoras para el desarrollo de software.

El presente estudio brindará información para la comunidad de desarrolladores de habla hispana interesada en Yii 2, ya que en la actualidad la documentación en español a nivel

mundial abarca el 3% según el número de búsquedas obtenidas de Google Search con el patrón de búsqueda: “Yii 2” Yii.

Este estudio también ayudará a Start-Ups¹⁸ y a los profesionales con sus proyectos de desarrollo de software, ya que brindará información sobre cómo hacer una aplicación empresarial con el framework.

En cuanto al aplicativo a desarrollar, este se enfoca en la fase 2 del sistema nutrimental infantil, al cual se agregarán funcionalidades avanzadas a través de la seguridad basada en roles, la creación de preparación de platillos y la generación de bitácoras del consumo de alimentación de los infantes.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. ESTADO DEL SEGUIMIENTO NUTRIMENTAL INFANTIL EN EL ECUADOR

Ecuador tiene planes para disminuir la desnutrición crónica en menores de cinco años. Dentro de esto se encuentra el plan “Acción Nutrición” el cual busca modificar los factores que determinan la malnutrición. Este plan se ejecuta con la ayuda conjunta del Ministerio de inclusión económica y social (Mies¹⁹), Instituto de economía popular y solidaría (Ieps²⁰), Ministerio de agricultura, ganadería, acuacultura y pesca (Magap²¹) para mejorar las dietas alimenticias infantiles que reciben los niños dentro de los Cibv.

Mientras que el MSP es quien vela por la salud integral infantil de los niños a través de las tomas de medidas antropométricas y revisiones periódicas de salud.

2.1.1. DEFINICIÓN DEL SEGUIMIENTO NUTRIMENTAL INFANTIL EN LOS CIBV

Los Cibv tienen como misión brindar una atención de calidad promoviendo el desarrollo infantil integral, el cual se menciona como: “un proceso educativo que propicie de manera equitativa e integrada el alcance de niveles de desarrollo en diferentes ámbitos físico, socio-afectivo, motriz, intelectual y de la comunicación y lenguaje”. La Dirección nacional de servicios de centros infantiles del buen vivir señala que un Cibv se constituye a partir de una población mínima de 40 infantes aplicando excepciones en zonas de alta dispersión o vulnerabilidad. Dentro de las actividades que los Cibv deben de cumplir se encuentran: el desarrollo integral, la alimentación, el descanso e higiene y la continuidad en el hogar. La salud de los infantes es asumida desde el punto de vista preventivo y de acuerdo a las normas de atención primaria de salud para la niñez del MSP y que los Cibv cubren con un 75% de las recomendaciones nutricionales diarias a través de los cuatro tiempos de comida: desayuno, refrigerios y almuerzo, el otro 25% es responsabilidad de la familia. (Dirección Nacional de Servicios Centros Infantiles del Buen Vivir, 2013) (págs. 4 - 5,11 ,18 - 19).

2.1.2. PROCEDIMIENTOS DE LAS ENTIDADES DE SALUD

El artículo 66 literal 2 de la Constitución del Ecuador del 2008, declara como derecho el tener una vida digna que asegure la salud, alimentación y nutrición, es por tal motivo que el Mies en su publicación “Norma Técnica de Desarrollo Infantil Integral” indica que: Los Cibv deben de realizar actividades de: controles de salud, valoración del estado nutricional, cumplimiento del esquema de vacunación, suplementación con micronutrientes, desparasitación a partir de los dos años y control de salud bucal de acuerdo a la periodicidad que impone el MSP (MIES, 2014, pág. 16).

En la norma técnica para los Cibv la alimentación infantil es una actividad garantizada y está enmarcada dentro de las recomendaciones nutricionales que el MSP utiliza, es así que con el consumo de una dieta balanceada se puede disminuir el porcentaje de población infantil malnutrida; en donde muestra que el principal indicador de desnutrición es la anemia que es efecto de la carencia de hierro en la alimentación y afecta al 60% de la población infantil menores de dos años. (Ministerio Coordinador de Desarrollo Social, 2015).

2.2. SITUACIÓN ACTUAL DE APLICACIONES EN YII 2

Debido a las altas métricas de rendimiento de la versión 1 de Yii, su integración con módulos, extensiones y componentes, la comunidad encargada del desarrollo del código base del sistema (core) en su versión 2 observó la necesidad de incorporar estándares de escritura enmarcados en la cooperación entre desarrolladores, estándares del lenguaje PHP y utilización de Composer para la integración y actualización de librerías dependientes del framework (Xue, Makarov, Brandt, Klimov, & Yii, 2014). Aunque el framework en su versión 2.0.0 fue lanzado en octubre del 2014, esta fue lanzada para la detección y corrección de errores, de este modo los usuarios de todo el mundo contribuyeron con mejoras hasta la actual versión 2.0.6 la cual es catalogada como estable.

Las tendencias de búsqueda de frameworks PHP se visualizan en la Figura 3, esta fue obtenida de la herramienta Google Trends y evidencia el crecimiento del interés en este framework debido a las prestaciones y fácil integración del mismo, así como del soporte y resolución de errores que brinda la comunidad de Yii 2 en internet a nivel mundial.

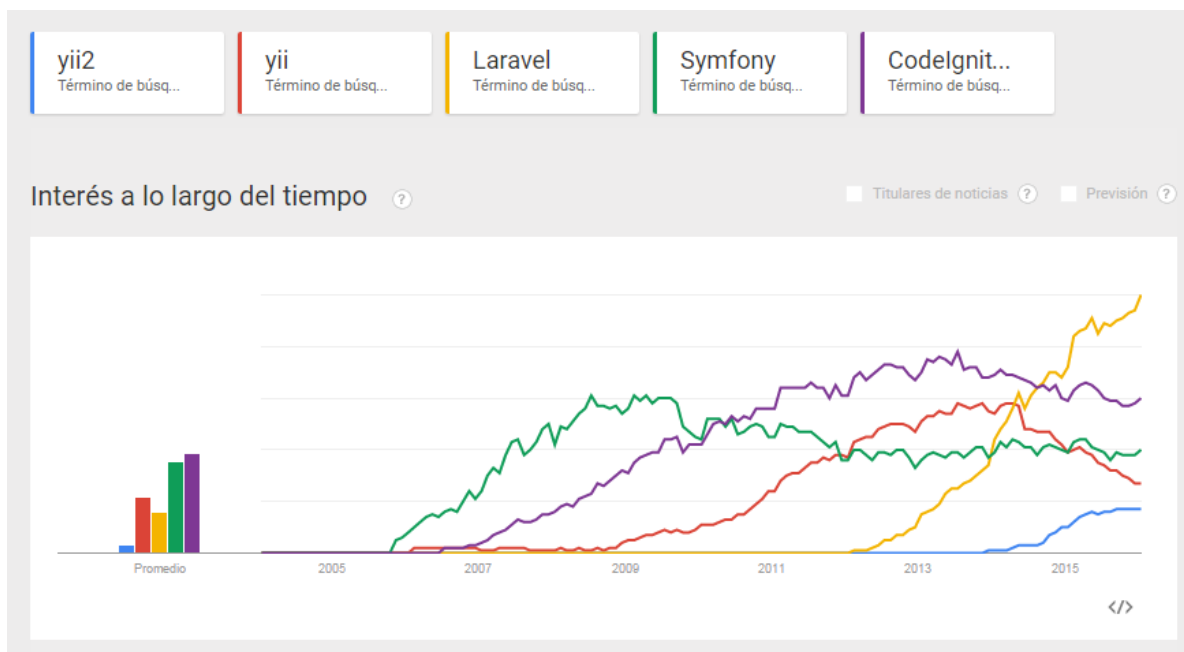


Figura 3: Búsqueda de frameworks PHP según Google Trends revisado el 29-01-2016

Fuente: Propio

La cantidad de información sobre el framework Yii 2 abarca un 97% en lenguas distintas al español de este modo la búsqueda de información de este framework en la población de habla hispana sin conocimientos de otros idiomas resulta difícil de realizar. La Figura 4 muestra la cantidad de información de la total mundial acerca del framework de estudio filtrado por idiomas.

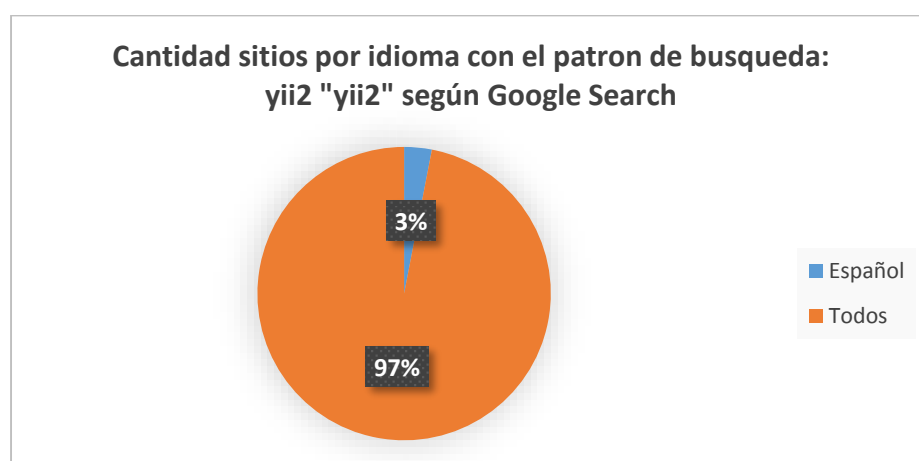


Figura 4: Porcentaje de información en Internet de Yii 2 según Google Search

Fuente: Propio

Actualmente algunas empresas de desarrollo en especial de la India, de donde es originario gran parte del código de Yii, optaron por Yii 2; ya que este promete evolucionar conforme se desarrollen nuevas actualizaciones del lenguaje PHP. En Ecuador el desarrollo de aplicaciones web con Yii 2 es una meta a lograr dentro de empresas que usan Yii 1 ya que este se acopla con metodologías de desarrollo ágiles como XP²² o Scrum²³.

DEFINICIÓN DEL MARCO DE YII 2

Yii proviene del vocablo chino Yee que significa “simple y revolucionario” pero por su pronunciación en el occidente fácilmente optó por ser nombrado como Yii que es el acrónimo de “Yes It Is” (Yii Software LLC, 2015).

Es un framework MVC²⁴ catalogado como software libre, orientado a objetos y de alto rendimiento basado en PHP para el desarrollo rápido de aplicaciones Web de gran escala debido a su alta extensibilidad y fácil ajuste para mejoras en rendimiento (Yii Software LLC, 2015). Yii 2 se acopla fácilmente a entornos de desarrollo, es por ello que esta versión de Yii trabaja con Composer. Cuando se instala una nueva aplicación esta tiene pre configurado los estándares de la estructura de directorio sea un Template básico o avanzado, además de tener activas algunas características básicas como: página de login, formulario de contacto, página de inicio y página de registro de usuarios.

2.2.1. CARACTERÍSTICAS DE YII 2

La versión 2 de Yii es un framework de programación genérico, lo que significa que este puede ser usado en todo tipo de aplicaciones web. Cuenta con una arquitectura basada en componentes y un soporte sofisticado de caching²⁵.

El framework implementa el patrón de diseño MVC, toma la filosofía de que el código debe de ser realizado una sola vez de forma elegante es decir entendible para otras personas. Fue construido para ser un framework full-stack²⁶ para proveer de constructores de consultas y ActiveRecord²⁷ para base de datos relacionales y no relaciones. Yii2 soporta de forma nativa el desarrollo de web services tipo RESTful y caching multi-tier. Tiene la característica de ser extremadamente extensible²⁸, logrando tomar ventaja de la arquitectura de Yii para distribuir o implementar extensiones de módulos o componentes. Está dividido en dos

Templates (básico y avanzado) según la complejidad, tamaño y requerimientos del proyecto (Xue, Makarov, Brandt, Klimov, & Yii, 2014).

2.2.1.1. BASIC TEMPLATE

El Template básico permite tener el código con una estructura sencilla, separa modelos, vistas y controladores, así como los recursos, extensiones y widgets propios del framework o de terceros dentro de una carpeta denominada vendor. Tiene la capacidad de separar el código en módulos y trae por defecto el generador de código Gii que realiza la mayor parte de codificación del framework traduciendo y generando código desde la base de datos a la arquitectura MVC. La configuración de los componentes, módulos y conexiones a la base de datos se lo realiza a través de los archivos de configuración en el apartado config y permite la configuración del componente de RBAC que trae Yii 2 en su código base.

2.2.1.2. ADVANCED TEMPLATE

Agrupar las características del Template básico en dos entornos: Frontend y Backend. De acuerdo a (Vahrmeijer, 2014) quien habla del uso del Template avanzado cuando se tiene dos o más bases de datos debido a que brinda la facilidad de separar y mantener la organización el código en una misma aplicación. Otra forma de entender el uso de este es tener el Frontend como parte de la aplicación la cual usaran los clientes, mientras que el Backend servirá para la administración de la propia página web. También se da a conocer que ayuda a los diseñadores junior a que prefieran esta separación debido a que no necesitan modificar tanto código del lado del servidor, separando el Frontend en plantillas de vista PHP con ficheros .css y .js mientras que el Backend tendrá los controladores y modelos.

Este Template es usado para solucionar requerimientos que necesiten de Web Services, separando este código en un tercer entorno denominado generalmente como API. Si se opta por el uso del Template avanzado se debería utilizar las recomendaciones de (Vahrmeijer, 2014) listadas a continuación:

- Usar la separación en Backend y Frontend únicamente si los requerimientos lo hacen una necesidad.
- Para aplicaciones que requieren alta seguridad no se recomienda mezclar Frontend y Backend; se debería tener paneles, tablas o base de datos distintas
- Se puede compartir componentes como por ejemplo el RBAC, widgets y modelos entre Frontend y Backend cuando la seguridad o los requerimientos del sistema lo requieran.
- Es fácil de dar mantenimiento, aunque la complejidad de la programación y estructura de directorios aumenta.
- La configuración del Template avanzado puede ser de gran utilidad conforme el desarrollador ya tenga experiencia en el área.

2.2.2. CICLO DE VIDA DE LAS PETICIONES

Cada vez que una petición es realizada por un usuario, Yii 2 maneja el flujo de acciones que se muestra en la Figura 5:

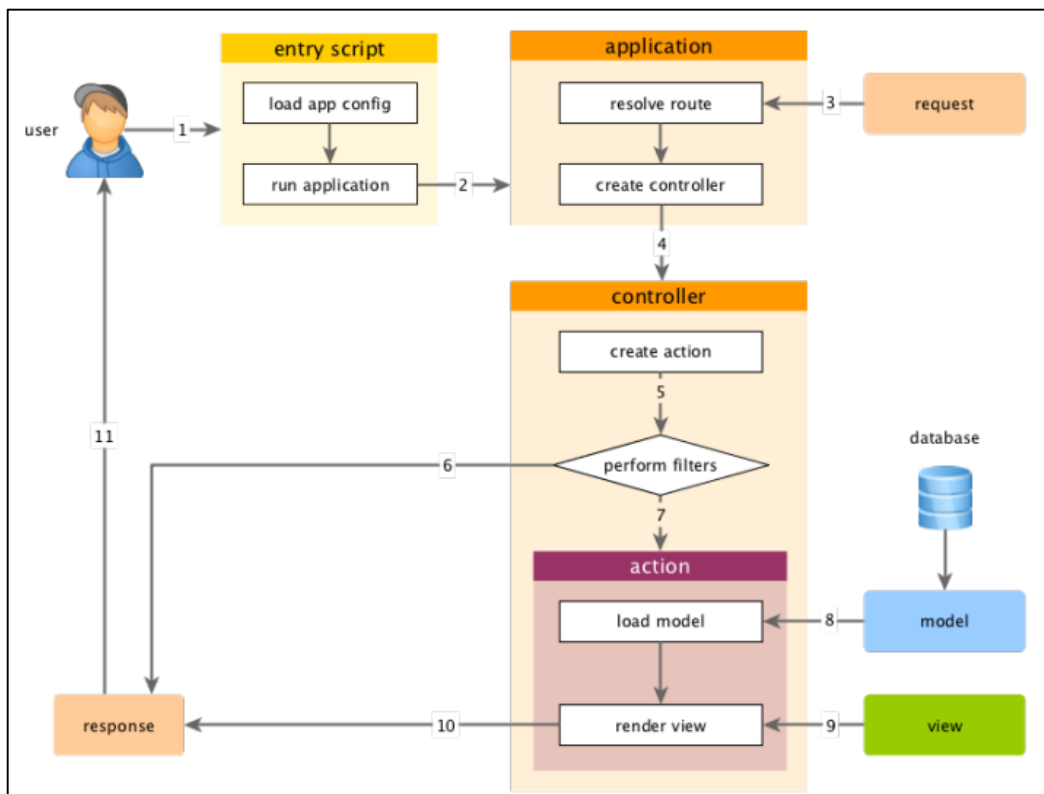


Figura 5: Flujo de la gestión de una aplicación

Fuente: Yii 2-guide

Los pasos descritos a continuación son tomados de (Xue, Makarov, Brandt, Klimov, & Yii, 2014) y explican el ciclo de vida de las peticiones en Yii 2.

Paso 1: Un usuario realiza una petición al script de entrada index.php

Cualquier petición que un usuario realiza a una aplicación en Yii 2 se comunica con el fichero index.php o script de entrada.

Paso 2: El script de entrada carga las configuraciones de la aplicación y crea una instancia para manejar la petición.

El fichero index.php tiene las instrucciones necesarias para poder buscar y cargar las configuraciones de la aplicación. Una vez realizado este procedimiento se procede a instanciar una nueva aplicación con los datos de las configuraciones.

Paso 3: La aplicación resuelve la ruta solicitada con la ayuda del componente Request de la aplicación.

El componente Request²⁹ crear un objeto capaz de manejar la información y características de las peticiones para poder posteriormente en los controladores específicos, analizar y re direccionar dicha petición enviada por el usuario.

Paso 4: La aplicación crea una instancia de un controlador para manejar la petición.

Con los datos de la petición en el paso 3 se puede direccionar al fichero de un controlador para poder crearlo. El core de Yii 2 provee de un componente denominado URLManager³⁰ para conocer cuál es el fichero correcto al cual direccionar.

Paso 5: El controlador crea una instancia de una acción y ejecuta los filtros de dicha acción.

Del mismo modo que se conoce cual fichero es requerido a través de la petición, el componente URLManager indica a que acción de dicho controlador debe ser creada y ejecutada. Dentro de los filtros que pueden afectar a las acciones se encuentran los de control de acceso para la validación de usuarios y tipos de peticiones.

Paso 6: No pasa las validaciones

Si alguno de los filtros falla la acción es cancelada y se envían notificaciones de error en el componente Response³¹.

Paso 7: Pasa las validaciones

La acción es ejecutada si todos los filtros pasan.

Paso 8: La acción carga datos del modelo, posiblemente de la base de datos.

Las instrucciones dentro de la acción generalmente llaman a los modelos de la aplicación para poder usar los datos existentes en una base de datos u otro tipo de contenedor de información. Yii 2 provee de un conjunto de objetos para la comunicación mediante un ORM³² a la base de datos, entre ellos el más usado es el ActiveRecord.

Paso 9: La acción renderiza una vista, pasándole los datos del modelo cargado.

La acción del controlador ubica el fichero de vista y a través de métodos de renderización inyecta los datos obtenidos del modelo en este, el resultado de este procedimiento es un objeto listo para ser enviado al navegador con la información que se pidió según la ruta de la petición en el paso 1.

Paso 10: El resultado de la renderización es pasado al componente response de la aplicación.

Cuando la aplicación termina los procesos con la petición, se genera un objeto Response, el cual es enviado al usuario final. Este es generado desde la clase `yii\web\Response` y contiene información de los códigos de status, cabeceras y cuerpo del HTTP.

Paso 11: El componente response envía el resultado de la renderización al navegador del usuario.

El navegador recibe la respuesta con la información solicitada en la petición del paso 1.

En el anexo A se describe el flujo de peticiones de una aplicación desarrollada en Yii 2.

2.2.3. VENTAJAS DE DESARROLLAR CON YII 2

Yii 2 posee las ventajas siguientes:

- Implementa el uso de Composer que permite montar un ambiente de desarrollo fácil y resuelve las dependencias de librerías de terceros de forma automática.

- Tiene una estructura de directorios que mejora la seguridad, ya que anteriormente se poseía únicamente la seguridad de la configuración del archivo `.htaccess`³³ para que los usuario accedan a los archivos de recursos web.
- Aprovecha las características avanzadas de PHP como: enlace estático en tiempo de ejecución³⁴, usa SPL³⁵ o librerías de estándar PHP y funciones anónimas³⁶.
- Usa namespaces³⁷ permitiendo que el mecanismo de auto carga de clases sea más efectivo al momento de localizar e incluir todos los requerimientos de archivos.
- Usa alias para definir rutas de archivos o URLs para evitar escribir estas y que el código sea más pulcro.

2.3. PATRÓN DE DISEÑO MVC

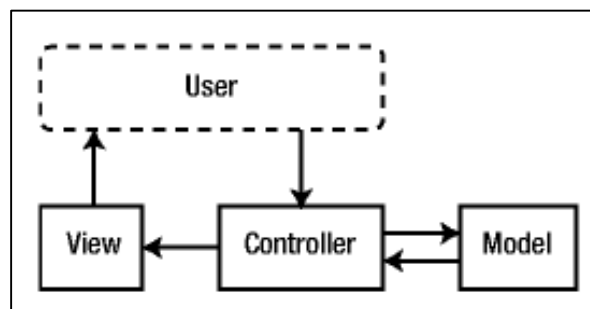


Figura 6: Esquema de MVC

Fuente: (Pitt, 2012)

Según (Pitt, 2012), el MVC es un patrón de diseño construido para interconectar tres componentes escritos en un paradigma de programación orientada a objetos. Siendo estos modelos, vistas y controladores. Este diseño está construido uniendo otros patrones diseño, entre ellos se encuentran: Singleton y Factory de los cuales Yii 2 también saca provecho.

2.3.1. MODELO

El modelo contiene la lógica del negocio de una aplicación, por ejemplo, los métodos de cómo se almacena la información de una aplicación o las políticas y validaciones para poder cumplir con los requisitos de negocio.

2.3.2. VISTA

La vista contiene las presentaciones de la interfaz con las que el usuario interactúa con el sistema, estas pueden incluir HTML, CSS y archivos Javascript. Muchas veces lo que ve el

usuario es en realidad una combinación de muchas de vistas diferentes las que viajan por una misma petición.

2.3.3.CONTROLADOR

El controlador es el componente que conecta los modelos y vistas. Estos aíslan la lógica de negocio de un modelo a partir de lo que se requiera en la interfaz de una vista del usuario.

Los controladores son el primer punto de entrada de las peticiones de entre los tres componentes, debido a que la solicitud pasa primero al controlador instanciando los modelos y vistas necesarias para cumplir con la petición del usuario final.

2.4. MVC EN YII 2

La estructura de aplicaciones en Yii 2 puede ser entendida con la Figura 7, en donde se muestra las características que posee el framework conjuntamente con el patrón de diseño Modelo Vista Controlador.

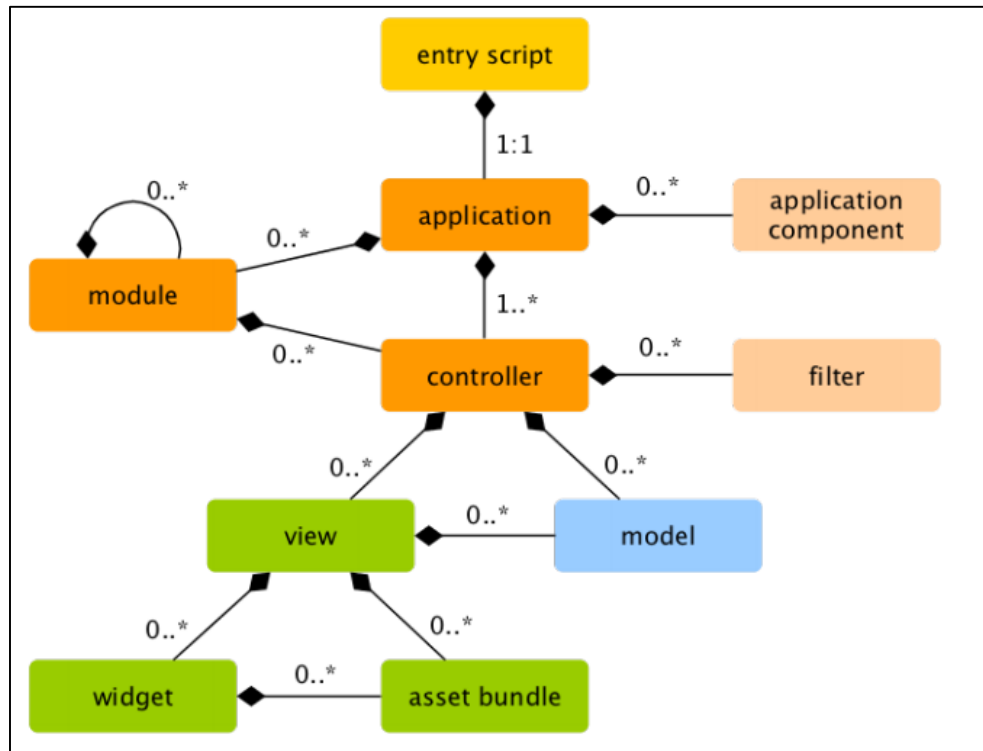


Figura 7: Esquema MVC Yii 2

Fuente: (Yii Suporters, 2015)

La Figura 7 es explicada a través de la Tabla 1:

Tabla 1:

Descripción de la arquitectura de Yii 2 con MVC

Objeto	Descripción
Entry script	También llamado script de entrada, es el primer eslabón en el proceso de bootstrapping ³⁸ de la aplicación. Una aplicación tiene un único script de entrada.
Application	Las aplicaciones son objetos que gobiernan la estructura general y ciclo de vida de aplicaciones echas en Yii.
Application component	Los componentes de aplicación son objetos que proveen de servicios para el procesamiento de la petición, entre los servicios que se tiene están: componente URLManager ³⁹ , componente db ⁴⁰ , Request, Response, etc.
Filter	Los filtros son objetos que se ejecutan antes y/o después de las acciones de los controladores.
Module	Los módulos son unidades independientes de software con modelos, vistas, controladores y componentes, dependen de una aplicación en donde residir para poder ser usados.
Controlador	Objetos que sirven de puente entre los modelos y las vistas
View o vista	Presentan la información de los modelos en un formato para interacción de los usuarios finales.
Widget	Pequeños bloques de código para las vistas, estos son reusables y ayudan a mejorar la interfaz del usuario
Asset bundle	Colección de archivos de assets dentro de un directorio, estos permiten administrar y configurar eficientemente estos ficheros.
Modelo	Representación de la información que se obtiene de colecciones como bases de datos, ficheros, etc.

Fuente: (Yii Suporters, 2015)

2.4.1.VISTA EN YII 2

Las vistas son parte de la arquitectura MVC, este código es el responsable de presentar los datos a los usuarios finales. Existen Templates para las vistas las cuales dan los términos de la composición del código HTML y del código PHP para la presentación. En Yii 2 estos son administrados por el componente de aplicación `yii\web\View` el cual provee de métodos para facilitar la composición y renderización (Xue, Makarov, Brandt, Klimov, & Yii, 2014) .

Una vista es un script PHP mezclado con código HTML y PHP. El código PHP es usado para generar el contenido dinámico como títulos, inputs y demás, mientras que el código HTML organiza el código de la página. Dentro de estas vistas, existen variables para manejar y colocar los datos del modelo a través de los controladores en la capa de vistas.

Seguridad en las vistas

Yii provee de seguridad contra ataques de cross-site scripting⁴¹. Posee dos funciones ambas se heredan de los helpers de yii, `Html::encode()` permite codificar texto plano y `HtmlPurifier::process()` filtra el contenido y luego muestra en pantalla (Xue, Makarov, Brandt, Klimov, & Yii, 2014).

2.4.1.1. ESTRUCTURA DE LAS VISTAS

La Figura muestra cómo se conforman los distintos tipos de objetos de vistas dentro del framework.

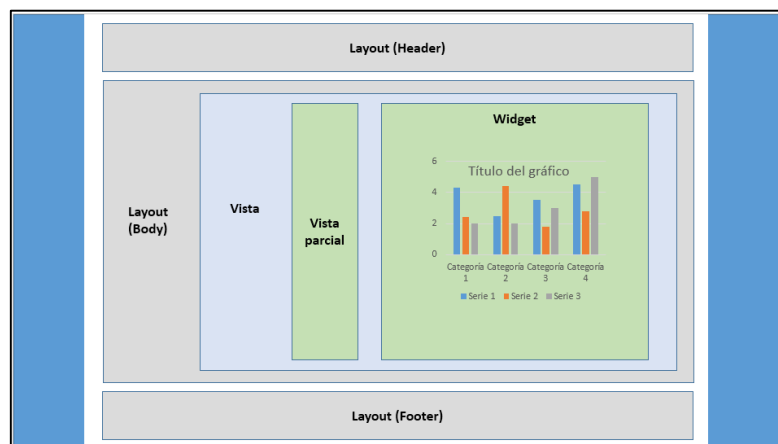


Figura 8: Esquema de vistas Yii 2

Fuente: Propio

Layouts

Los layouts son un tipo especial de vista que representan partes comunes de múltiples vistas como la sección de cabecera (header) que puede contener menús (ítems de navegación), el pie (footer) que puede contener información extra sobre contactos y el cuerpo donde se inserta el contenido dinámico de la aplicación (Xue, Makarov, Brandt, Klimov, & Yii, 2014).

Yii 2 trabaja con layouts y los tiene almacenados por defecto en el directorio `@app/views/layouts`. Estos tipos especiales de vista generan etiquetas HTML que son comunes para todas las vistas como se presenta como el esquema del layout presentado en la Figura 9.

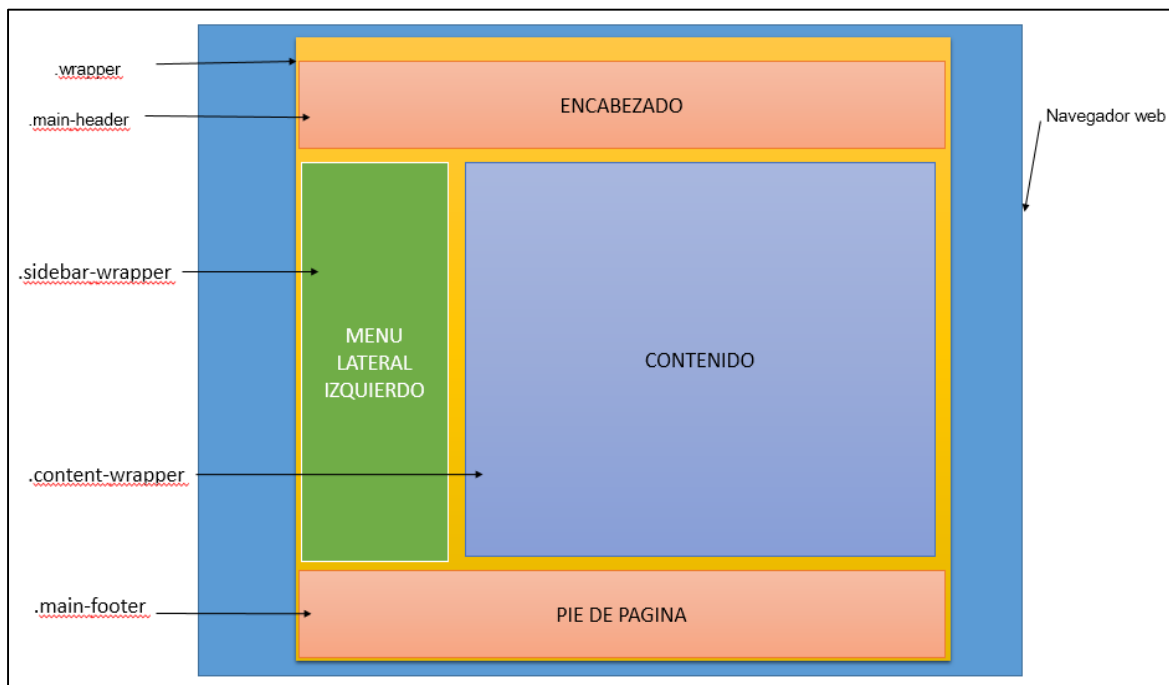


Figura 9: Esquema del layout del theme AdmLte

Fuente: Propio

En Yii 2 los layouts se componen de eventos para el proceso de renderización de scripts y etiquetas registrados en otros lugares, es decir realiza una inyección de código dentro del esquema del layout. Los eventos del layout controlan los lugares donde se inyectarán porciones de código como etiquetas, cuerpo de la página, cabecera y pie de esta forma el layout se torna en un objeto totalmente configurable. La activación de un layout se lo realiza a través del atributo `$layout` desde un controlador.

Vistas y vistas parciales

Las vistas son ficheros que contienen lo necesario para visualizar información y están formados por porciones de código HTML y PHP. Estas son inyectadas dentro del layout antes de su renderización y consisten básicamente en elementos de la interfaz de usuario (user interface - UI). Las vistas parciales al igual que las vistas son porciones de código HTML y PHP que sirven de contenedor para visualizar información específica, por ejemplo: gráficos, imágenes, menús, etc. Estas son renderizadas dentro de las vistas y sirven para separar la estructura del código en la vista (Yii-Suporters, 2016).

Widgets

Son elementos de una interfaz gráfica y basan su funcionalidad en proveer un punto de interacción con el usuario para la manipulación directa de un tipo de dato dado, son usualmente embebidos en los scripts de vista para generar interfaces de usuarios complejas ya que permite la interacción de este con eventos que el usuario realice (ALEGSA, 2016).

Assets

En Yii 2 un asset es un archivo o recurso que se enlaza o se referencia con una página web, como: .js, .css, imagen, audio y video. Automáticamente el sistema incluye los assets necesarios para el funcionamiento, en lugar de mandar a pedir y posteriormente incluir estos. El framework organiza y administra los assets de tal modo que se podrán usar y cargar en el navegador en una sección diferente en la estructura del proyecto. Por lo general, los assets necesarios para el correcto funcionamiento de un Widget se encuentran en el directorio bower, a estos assets se los puede denominar como assets de fuente. Estos recursos sirven de fuente para generar ficheros en la carpeta de publicaciones ubicada dentro de (frontend|backend) /web/assets los cuales son consumidos por los navegadores y dispositivos web (Xue, Makarov, Brandt, Klimov, & Yii, 2014).

Assets Bundle

Es una colección de assets localizados en un directorio, cuando se registre un asset Bundle en una vista, este incluirá los archivos .css y .js en el Bundle (paquete) en la página web renderizada. (Xue, Makarov, Brandt, Klimov, & Yii, 2014). El asset Bundle es un archivo de php que extiende de la clase yii\web\AssetBundle.

Si se desea que se carguen archivos en un orden en específico, esto se lo debe de configurar en la sección de depends, ya que se cargará n en orden descendente desde el primer Asset que se encuentre en ese vector, dejando al final los assets de nuestro paquete.

2.4.1.2. THEMES

Según (Xue, Makarov, Brandt, Klimov, & Yii, 2014) un theme es un conjunto de vistas y archivos de layout, cada theme puede proveer distintas experiencias al usuario final por lo que muchos themes pueden ser usados en una misma aplicación.

EL theme que se presenta en la Figura 10 se lo integró al Frontend de la aplicación a desarrollar. Este tiene componentes y un diseño más acorde a las necesidades de los usuarios finales, es decir sin característica de administración.

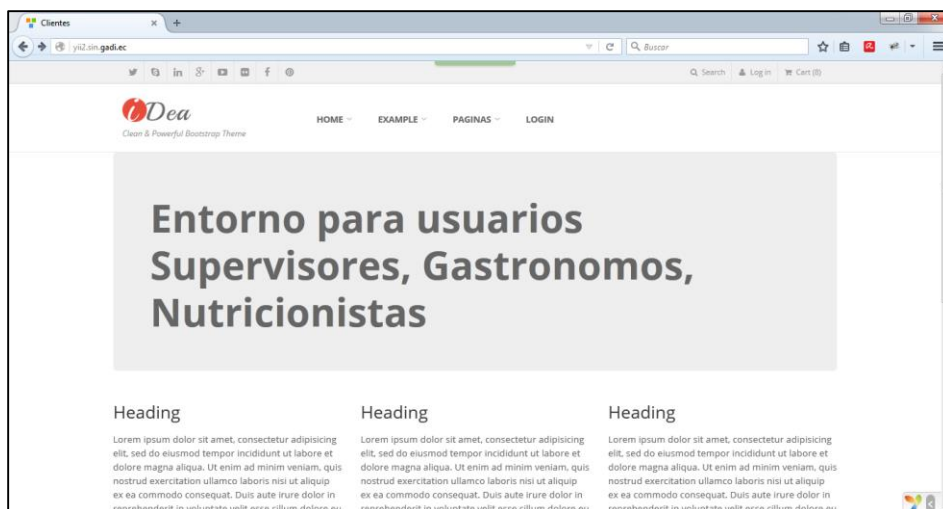


Figura 10: Theme idea_v1.3 integrado al framework

Fuente: Propio

EL anexo A muestra los pasos necesarios para la instalación y configuración de un theme.

2.4.2. MODELO EN YII 2

El modelo se levanta sobre una librería de abstracción de acceso (PDO⁴²) y se define como una interfaz para manipular las bases de datos. (Popel, 2007). Yii 2 soporta las siguientes bases de datos: MySQL, MariaDB, SQLite, PostgreSQL, CUBRID, Oracle y MSSQL.

Los parámetros de configuración para las bases de datos se encuentran en el objeto db, este se encuentra por defecto en los archivos config/main.php del Frontend, Backend y Common en un Template avanzado y en el fichero config/db.php con el Template básico.

Gestión de base de datos dentro de Yii 2

Yii 2 provee de características avanzadas que son fácilmente configurables a través del fichero de configuración comúnmente nombrado config.php. Se tiene las siguientes características:

- Es posible la configuración del control de los niveles de aislamiento de las transacciones, por defecto se encuentra con el nivel de aislamiento que posea la base de datos.
- El componente de conexión que se encuentra en yii\db\Connection soporta configuración para replicación de base de datos, comportamiento de Read-Write Splitting⁴³, balanceo de carga y comportamientos de conmutación de error de los esclavos (failover between slaves⁴⁴).

2.4.2.1. OBJETOS DE ACCESO A LA BASE DE DATOS (DAO⁴⁵)

Yii 2 tiene una capa construida sobre los objetos de datos de PHP (PDO) denominada DAO. Esta capa provee de una amplia API⁴⁶ y resuelve algunas inconsistencias entre las distintas bases de datos. Sobre esta capa se crearon los demás métodos de acceso a datos como el Query Builder y Active Record. Cuando se usa directamente el DAO es necesario utilizar

sentencias SQL y arrays de PHP. El DAO que provee Yii 2 también permite el trabajo con bases de datos NoSql (Xue, Makarov, Brandt, Klimov, & Yii, 2014).

El DAO de Yii 2 provee de métodos para manipular los esquemas de las distintas bases de datos como creación, eliminación, actualización de tablas, columnas, claves primarias, claves foráneas e índices. Es la forma más eficiente de acceso a la base de datos, pero debido a las distintas sintaxis de SQL esta requiere de más esfuerzo al realizarlo con bases de datos NoSQL.

En el anexo B sección B.3 y B.4 se muestra el uso del DAO en Yii 2.

2.4.2.2. QUERY BUILDER

Se encuentra sobre el DAO y permite construir declaraciones de SQL de forma programada más legibles y seguras. El Query Builder es un objeto del core de Yii 2 y representa las distintas partes de una declaración SQL. Este objeto constructor de consultas permite obtener consultas entendibles en forma de objetos ya que soporta al lenguaje de definición de datos (DDL) y manipulación de datos (DML) como métodos en su estructura. Se compara a continuación el uso del Query Builder vs una declaración en SQL, el primero organiza y estructura a través de arrays y métodos una sentencia SQL, lo cual mejora la legibilidad sobre los datos que se requieren (Yii Suporters, 2015).

```
$rows = (new \yii\db\Query())
->select(['id', 'email'])
->from('user')
->where(['last_name' => 'Smith'])
->limit(10)
->all();

SELECT `id`, `email`
FROM `user`
WHERE `last_name` = :last_name
LIMIT 10
```

Figura 11: Uso de Query Builder y declaración generada por el mismo

Fuente: (Yii Suporters, 2015)

En el anexo B sección B.5 se describe el uso del Query Builder.

2.4.2.3. ACTIVE RECORD ORM

Es una interfaz orientada a objetos para acceder y manipular las bases de datos. Se caracteriza por que asocia una clase Active Record a una tabla de base datos, una instancia de esta clase corresponde a una fila de la tabla y maneja los atributos de la clase como representación de una columna de una tabla. Este interfaz permite la persistencia con las clases del modelo, su complejidad es mayor que la del Query Builder y esta provee de métodos Get que sirven para la comunicación entre los demás modelos haciendo símil a las relaciones que tienen las tablas en el diagrama entidad relación. La Figura 12 muestra la diferencia entre los métodos que provee el DAO y Active Record en donde se realiza el ingreso de un registro, se deduce que Active Record es más fácil, entendible y pulcro al realizar operaciones CRUD⁴⁷ mientras que al usar el DAO se genera más código.

```
$customer = new Customer();
$customer->name = 'Qiang';
$customer->save();

$db->createCommand('INSERT INTO `customer` (`name`) VALUES (:name)', [
    ':name' => 'Qiang',
])->execute();
```

Figura 12: Comparación entre Active Record y DAO para la inserción de un nuevo registro en la tabla Customer. Superior: Uso de Active Record. Inferior: Uso de DAO.

Fuente: (Yii Suporters, 2015)

En el anexo B sección B.6 se describe el uso del Active Record.

Formas de acceso a tablas relacionadas

Active Record permite acceder a los datos de tablas relacionadas y devuelve un conjunto de datos en términos de un array Active Record de la clase relacionada. Estas relaciones se crean a través de métodos Get que apuntan a las otras tablas con las que se tiene relación en el modelo de entidad relación. La Figura 13 muestra una relación entre las tablas order y

customer mientras que la Figura 14 traduce esta relación a código para su posterior uso en el MVC.

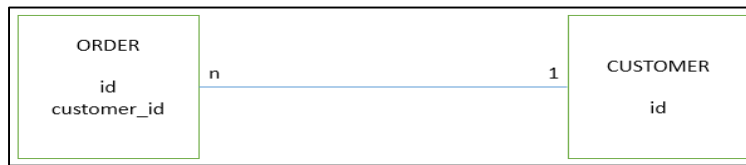


Figura 13: Modelo relacional one-to-many de ejemplo

Fuente: Propio

```
class Customer extends ActiveRecord
{
    public function getOrders()
    {
        return $this->hasMany(Order::className(), ['customer_id' => 'id']);
    }
}

class Order extends ActiveRecord
{
    public function getCustomer()
    {
        return $this->hasOne(Customer::className(), ['id' => 'customer_id']);
    }
}
```

Figura 14: Relaciones que se crean en los modelos de ejemplo

Fuente: (Yii Suporters, 2015)

Esta propiedad de relación funciona aun así en la base de datos no exista relaciones, por lo que ActiveRecord ayuda en gran medida cuando se desea crear relaciones “virtuales” es decir que no se visualicen en el diagrama de entidad relación. Del mismo modo que una consulta para la búsqueda de datos con clave foránea en dos tablas, se tiene como requisito al momento de crear estas propiedades, la multiplicidad de la relación siendo el método `hasMany()` y `hasOne()` los encargados de esta directriz, el otro modelo del active record también es necesario e indicar que atributos entra ambos modelos representan a la relación de multiplicidad.

La siguiente Figura 15 muestra una manera de enlazar los datos de tablas relacionadas, primero se accede a la tabla Customer la cual mediante la propiedad `getOrders()` tiene acceso a la tabla Orders y esta a su vez mediante la propiedad `getItems()` tiene acceso a los Items.

```
// eager loading "orders" and the nested relation "orders.items"
$customers = Customer::find()->with('orders.items')->all();
```

Figura 15: Uso de relaciones anidadas

Fuente: (Yii Suporters, 2015)

Métodos de acceso a datos: Lazy Loading y Eager Loading

La propiedad de relación de una instancia Active Record se puede acceder como una propiedad normal de un objeto, debido a esta característica es posible traer una cantidad gigantesca de información la cual puede llegar a saturar al servidor es por eso que los métodos Lazy Loading y Eager Loading dan control sobre el acceso a los datos relacionales y se caracterizan según lo indicado en la Tabla 2.

Tabla 2:

Formas de acceso (Lazy y Eager Loading)

Método de acceso	Explicación
Lazy Loading	Carga los datos relacionados al Active Record una única vez. Si se requiere acceder a un número considerable de datos de manera individual a través de un ciclo, realizara el número de peticiones que el ciclo indique.
Eager Loading	Permite reducir el número de peticiones que se realizan a la bdd cuando se requiere el acceso individual a un gran número de datos.

Fuente: Propio

2.4.2.4. ESTRUCTURA DEL MODELO

De acuerdo a la complejidad del objeto, se puede estructurar la capa de modelo en Yii 2 de la siguiente forma: en la cima se encuentra el ActiveRecord que llega a ser la clase de la cual extienden todos los modelos que tengan persistencia con la base de datos, la clase ActiveQuery sirve de ayuda al ActiveRecord para dinamizar las consultas de forma pragmática. Bajo esta se encuentra el objeto Query Builder que conforma parte del DAO el cual sirve para crear sentencias SQL de forma pragmática. El DAO posee una clase denominada Command que permite ejecutar código SQL directamente sobre la base de datos y está construida para dar persistencia a través del PDO a este y los demás objetos de acceso a datos del framework.

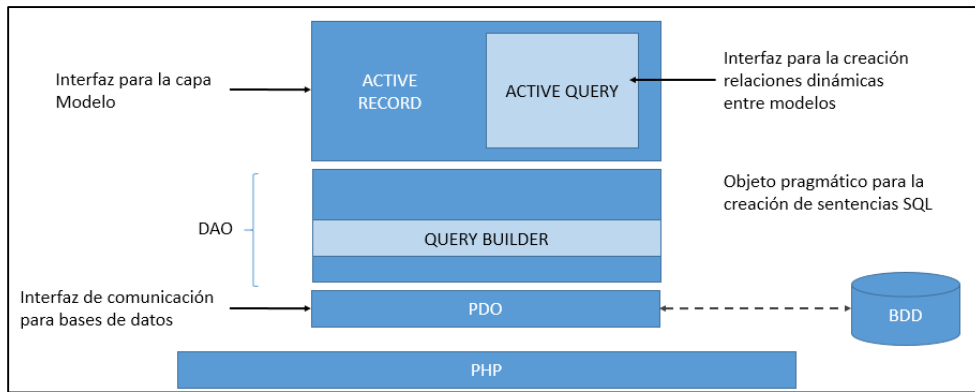


Figura 16: Objetos para gestión del modelo

Fuente: Propio

2.4.3. CONTROLADORES Y ENRUTAMIENTO EN YII 2

Los controladores son objetos de clases que extienden de `yii\base\Controller` y son los responsables del procesamiento de peticiones (Request) y la generación de respuestas (Response). Estos analizan las peticiones, pasan estas a los modelos, inyectan el modelo resultante en las vistas y finalmente generan las respuestas salientes. (Yii Suporters, 2015). Los controladores manejan peticiones con respecto a obtener recursos, por esta razón a menudo se les ubica por un sustantivo (ID) refiriéndose al tipo de recurso que maneja.

2.4.3.1. ACCIONES

Las acciones son la unidad más básica a la que los usuarios pueden dirigirse y solicitar para la ejecución de operaciones. Son métodos que pueden manejar las peticiones, respuestas, el acceso al modelo y las validaciones tanto del Request como de la sesión en el servidor. Deben de nombrarse siempre con el prefijo “action” seguido del nombre que deseamos darle en camel-case.

Tipos de acciones

Yii 2 define dos tipos de acciones: Inline Actions y Standalone Actions. Las acciones tipo Inline son aquellas que se refieren en términos de parámetros en el método mientras que las acciones Standalone se refieren a una clase de tipo Action.

Métodos de renderización

Dentro de los controladores se llama a métodos para renderizar vistas, estos métodos generan el contenido de la capa de vista necesario para ser presentados al usuario final.

Tabla 3:

Métodos de renderización en un controlador

Método	Explicación
render()	Renderiza a la vista nombrada y le aplica el layout.
renderPartial()	Renderiza la vista nombrada sin ningún layout.
renderAjax()	Renderiza la vista nombrada sin layout, e inyecta todos los scripts y archivos JS/CSS registrados. Esto sucede usualmente en respuestas a llamadas a AJAX Request.
renderFile()	Renderiza la vista especificada en términos de la ruta al archivo o alias.

Fuente: (Yii Suporters, 2015)

2.4.4.VALIDACIONES DEL MVC EN YII 2

Yii provee distintas fases para la validación de datos en las distintas capas del MVC. En el modelo se usa la función `rules()` para validar los atributos que provienen desde los formularios en la vista y son seguros de procesar, esta es la última fase de validación por la cual tienen que pasar los datos hasta ser persistentes en la base de datos, también existen validaciones desde el core de Yii las cuales vienen pre configuradas para los distintos tipos de datos, estos se encuentran en `yii\validators`.

En el controlador es recomendable validar las características de los objetos Request y Response, la validación de los filtros de controles de acceso AFC y la permisología del RBAC; Los filtros de control de acceso son sobrescritos a través del método `behaviors` de cada controlador y filtra únicamente a usuarios invitados de los autenticados, la validación de la permisología para los usuarios se lo realiza dentro de cada acción a través de la función `can()` del RBAC y la validación del Request y Response se lo realiza a través de comparaciones con la ayuda de métodos que estos objetos traen.

La regla de negación al acceso de las acciones del sistema se hace con el control de filtros de acceso (ACF) que Yii 2 trae por defecto, este describe a los usuarios invitados a través de la expresión '?'; mientras que los usuarios que están autenticados en el sistema son identificados con la expresión de '@' y tendrán acceso a todas acciones (Xue, Makarov, Brandt, Klimov, & Yii, 2014).

En la vista se puede realizar validaciones a través de los métodos que proveen los widgets y son las primeras en efectuarse, estos generan código HTML5 con las características de validación propias del lenguaje. Yii 2 genera automáticamente scripts Javascript para la validación de datos en los formularios, estas porciones de código se crean cuando se realizan los eventos de renderización de las vistas y obtienen sus directrices del método rules() del modelo para su configuración, también estas validaciones funcionan de forma asíncrona con ajax⁴⁸.

2.5. COMPONENTES DE APLICACIÓN DE YII 2

2.5.1. COMPONENTES

Todos los componentes son llamados mediante un objeto localizador de servicios, este objeto conoce como proporcionar los distintos tipos de servicios que la aplicación necesite, instanciando al componente usando su ID. Este se instancia desde yii\di\ServiceLocator. (Yii Suporters, 2015). Los componentes se comportan como variables globales por lo que sobrecargar la aplicación con componentes puede afectar al rendimiento del servidor. Yii 2 clasifica a los componentes en dos:

Componentes de aplicación: Son instanciados una única vez cuando son accedidos por primera vez, a partir de allí la aplicación usa la misma instancia. Tienen IDs predefinidos por el core y se los describe en la Tabla 4.

Componentes de bootstrapping: Permiten ser usados sin que sean accedidos instanciados y duran con la ejecución de cada petición dentro de este tipo de componentes se tiene al Log que permite gestionar de mensajes sobre eventos en la aplicación como errores, información o peligros con cada ejecución de una petición.

Tabla 4:
Componentes del framework Yii 2

Componente	Descripción
assetManager	Administra los paquetes de assets y su publicación.
Db	Representa una conexión a la base de datos.
errorHandler	Este componente maneja errores y excepciones de PHP, esta interfaz permite que no existan errores fatales de PHP sin tener un atrapador de excepciones (try and catch), además que permite mostrar el backtracking del error en ambiente de producción.
i18n	Este componente permite relacionarse con características de internacionalización (I18N) y localización (L10N).
mail	Soporta composición y envío de emails ya que el framework provee de una interfaz para esto, por lo general usa la extensión de yii2-swiftmailer.
response	Maneja la información que se envía al usuario final, es creada cuando una aplicación finaliza los procedimientos con el Request y contiene información del código de status, cabeceras y cuerpo del HTTP.
request	Representa la petición que llega desde el usuario y son objetos que proveen de información de los parámetros del request, Yii 2 provee formas para el manejo según el método de la petición, detección de urls y manejo de las cabeceras HTTP.
session	Representa la información que se maneja dentro de las sesiones de usuario en las aplicaciones web, por defecto Yii 2 abre una sesión por cada cliente que se conecta a la aplicación.
urlManager	Maneja las traducciones de peticiones HTTP y la creación de direcciones URL en base a un conjunto de reglas.
User	Representa la información de autenticación del usuario.
View	Soporta el renderizado de la vista, los themes, registro y renderización de scripts .css y .js , motores de Templates y manejo de assets.

Fuente: (Xue, Makarov, Brandt, Klimov, & Yii, 2014)

2.5.2. GENERADOR DE CÓDIGO

Es una extensión de Yii 2 que provee de un generador de código orientado a web para la creación automática de modelos, vistas, controladores, módulos y extensiones. Gii viene precargada dentro del core del framework y para su uso se lo debe de registrar en el archivo de configuración. Es posible también crear plantillas personalizadas para la generación de código a partir de las directrices con las que se maneje un theme.

2.5.3. CONTROL DE ACCESO BASADO EN ROLES (RBAC)

Es una funcionalidad para proveer distinción entre las actividades y privilegios de los distintos roles de usuario de un sistema; se dispone de un rol administrador principal quien tiene permisos sobre todas acciones y seguridad del sistema, el administrador del sistema tiene los mismos privilegios que el administrador principal exceptuando las tareas de seguridad y el operador que será un administrador junior y realizará operaciones de copia de seguridad y gestión de dispositivos. Estos roles no son obligatorios de implementar y dependen de las necesidades de seguridad de la organización (Oracle, 2016).

2.5.3.1. RBAC EN YII 2

Yii 2 provee de dos métodos para la verificación de permisos de usuarios y roles. El método de ACF⁴⁹ es una implementación que se usa en aplicaciones que necesitan de un control de acceso simple como en páginas web de carácter informativo y de aplicaciones que necesiten verificar a usuarios invitados y autenticados, para su configuración se requiere modificar el método que describe el comportamiento de los controladores. El segundo método hace referencia al RBAC el cual es una implementación de jerarquías de permisos siendo la más básico la regla que representa restricciones a los roles y permisos, el permiso se lo declara en una acción en un controlador para permitir el consentimiento de su ejecución y el rol que es un conjunto de permisos (Yii Software LLC, 2015).

El uso de RBAC del core de Yii con lleva a la realización de este en dos pasos. El primero es levantar los datos de autorización del RBAC y lo segundo es usar los datos de autorización para autorizar el chequeo de acceso en los lugares donde es necesario.

2.6. CARACTERÍSTICAS WEB RESPONSIVE

Una página web que utilice el diseño responsivo o adaptativo es aquella que busca la correcta visualización de una misma página en distintos dispositivos. Esto se realiza mediante el redimensionamiento y colocación de los elementos web de forma que se adapten al ancho de la pantalla mejorando notablemente la experiencia del usuario.

BOOTSTRAP

Es un framework de presentación que usa HTML5, Css3 y JQuery para que las páginas web sean agradables y estilizadas hacia los usuarios, siempre da la prioridad al para al diseño Mobile.

Utiliza un sistema responsivo de grillas y Css para el redimensionamiento de los elementos web, además de proveer de plugins como modales, carrouseles, popovers, etc. Mejorando sustancialmente la experiencia del usuario (Aravind Shenoy, 2014).

JQUERY

Es una librería de Javascript que se acopla perfectamente con el diseño dinámico de páginas HTML. Utiliza mecanismos de localización de los elementos del DOM a través de la sintaxis que CSS dicta. Su principal propósito es facilitar el uso de Javascript en el sitio ya que provee de métodos que envuelven código Javascript. Esta librería también provee de funcionalidades con AJAX y DOM del HTML (Swedberg, 2013).

OTRAS TECNOLOGÍAS (PJAX)

Es un plugin de JQuery que usa AJAX y el método pushState⁵⁰. Trabaja a través de peticiones Ajax para reemplazar el contenido HTML indicado para esta operación y a su vez actualiza la URL usando pushState sin que exista recarga de diseño u otros recursos Css o Js, simulando una carga completa de la página (Wanstrath, 2016).

CAPÍTULO III

3. DESARROLLO DE LA FASE DOS DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL Y DOCUMENTACIÓN TÉCNICA DE YII 2

3.1. SITUACIÓN ACTUAL DEL SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE UNO

La fase uno elaborada por la Srta. Amanda Sofía Mejía Andrade de la UTN está realizada con el framework Yii 2, cuenta con un entorno para la administración del sistema (Backend) y un entorno Frontend el cual será dispuesto a los demás usuarios del sistema. Actualmente se encuentra con filtros de control de acceso básicos para usuarios invitados y usuarios registrados, lo cual provee parcialmente de la funcionalidad deseada. No provee de procesos para la generación de registros sobre la alimentación infantil lo que impide conocer con precisión sobre el consumo alimenticio. La información utilizada para generación de menús o cartas es basada en un compendio de comidas locales y extranjeras, pero no cuenta con preparaciones típicas o poco conocidas de las localidades.

3.2. ACUERDO MIES Y GAD

Para cumplir los estatutos del Código orgánico de organización territorial y descentralización el GAD -I en su convenio con el MIES (GAD-I, 2015), se comprometen a cooperar mutuamente con la finalidad de desarrollar el Proyecto desarrollo infantil en el cual dan atención y garantías de calidad a 21 Cibv de la ciudad de Ibarra dado esto en el Proceso de inclusión social y educativa del GAD-I se le adjudica el control, supervisión y mantenimiento de estas entidades educativas.

3.3. METODOLOGÍA XP

Yii 2 puede trabajar perfectamente con cualquier metodología de desarrollo de software ya que no está desarrollado sobre una en específico, sin embargo los desarrolladores expertos sugieren utilizar una metodología de desarrollo ágil como XP o SCRUM; en el presente trabajo se eligió la metodología XP, debido a que prioriza las tareas dando resultados directos

en el menor tiempo posible lo que permite al cliente familiarizarse con la herramienta y a los programadores permite que la depuración del sistema se realice con cada historia de usuario (Tangient-LLC, 2016).

La Figura 17 permite identificar las fases que la metodología XP proporciona para el desarrollo de proyectos, esta explica que cada iteración se debe de tener un demo como producto y debe de ser presentado al cliente con un plazo mínimo de 1 semana en cada iteración.

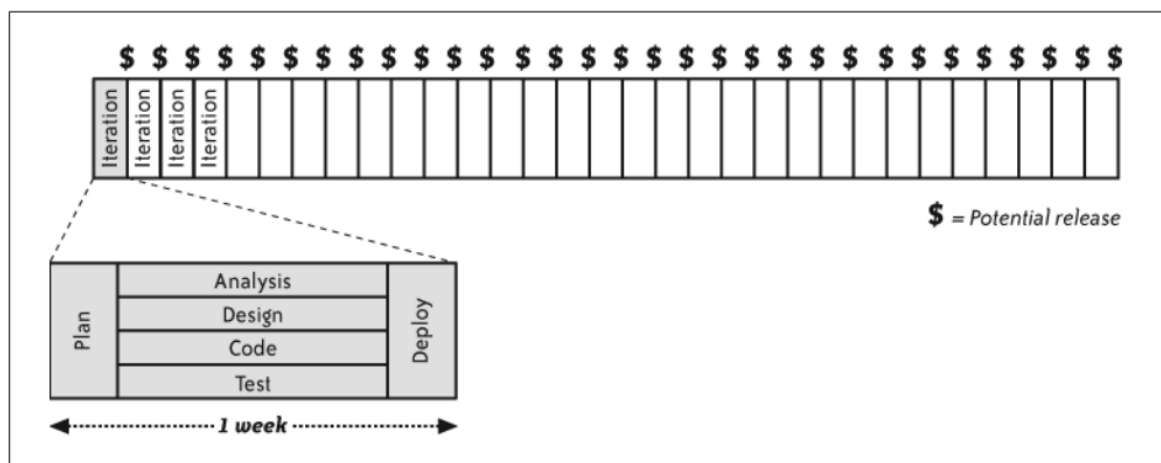


Figura 17: Ciclo de vida de XP

Fuente: (Shore & Warden, 2007)

3.3.1. PLANEACIÓN

En esta fase el equipo encargado de la metodología se debe reunir con los expertos de negocios y los clientes en donde se deberá de tomar las decisiones del negocio (objetivos y límites), se aclara la visión, fases del plan de lanzamiento y el manejo de riesgos del proyecto. (Shore & Warden, 2007). Se desarrollará historias de usuario y el plan de trabajo siguiendo la norma 830 de la IEEE⁵¹.

3.3.1.1. ANÁLISIS DE REQUERIMIENTOS

En la Norma técnica de desarrollo integral infantil (MIES, 2014) se describe los estándares y normativas para la prestación de servicios en los Cibv, es así que se describe a continuación los estándares que detallan los requerimientos de esta tesis.

Componente 3: Salud preventiva, alimentación y nutrición.

En el estándar número 16 fomenta las buenas prácticas de alimentación y promueve una alimentación saludable (MIES, 2014). En vista de esto desde el área de coordinación del desarrollo social del GAD-I se pidió el desarrollo de un módulo para poder incorporar y controlar las recetas de la localidad las cuales se incorporarán a planificaciones semanales de alimentación y poder así dar un seguimiento real de lo que consumen los infantes en los Cibv.

Componente 4: Talento Humano.

Este conjunto de estándares describe la conformación y funciones del equipo de talento humano que trabaja en los Cibv. Es por ello que se procederá a tomar los requerimientos de los roles de usuarios los cuales conformarán el módulo de RBAC.

La Dra. Susana Castillo Lara (usuario técnico de salud) en el desarrollo de la Fase 1 del SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL indicó que los Cibv deberían de controlar el desperdicio de comida y que se debería tomar notas acerca de cuál es el consumo real de los infantes ya que por hábitos alimenticios o por patologías estos no consumen en su totalidad la comida del Cibv, esta sugerencia fue presentada y aceptada por al área de coordinación del desarrollo social del GAD-I por lo que se procedió a planificar el módulo de bitácora de consumo.

3.3.2. DISEÑO

XP usa un diseño y estructura incremental para mejorar el diseño por etapas, se realizan pruebas que permiten la realización de tests con los diseños y la búsqueda de formas de refactorizarlos haciendo al código más limpio y funcional. (Shore & Warden, 2007) . Se procederá a crear los casos de uso de los diferentes actores del sistema en la fase dos, con esta información añadirá nuevas entidades al diagrama entidad relación para la base de datos y se creará los diagramas de los diferentes procesos que se realizaran en la fase dos.

3.3.3. CODIFICACIÓN

Cada iteración debe de ser implementada según lo planeado y se debe de supervisar a los stakeholders⁵² para la preparación del despliegue y resolución de problemas al final de una iteración antes de presentar con los clientes. Para poder mantener la consistencia e historial del código se usa software de control de versiones. (Shore & Warden, 2007). Es así que se procederá a implementar los requerimientos y cada fin de iteración se tendrá una versión presentable al cliente, en vista que el presente estudio es parte del Sistema de seguimiento nutrimental infantil se llevara el control de versionamiento con Git⁵³ en el servicio web que Bitbucket⁵⁴ ofrece a los equipos de desarrolladores.

3.3.4. PRUEBA

La fase de pruebas sirve para dar calidad a la aplicación, siendo los desarrolladores y clientes quienes se asegurarán de que la interfaz de usuario funcione según lo que requieren y finalmente el equipo de testers proveerá de calidad cuando la aplicación esté en producción. (Shore & Warden, 2007). Cada fin de iteración con el entregable se hará test con datos reales que ayudará a buscar errores y validar la programación y la recopilación de requerimientos.

3.4. DOCUMENTACIÓN TÉCNICA DE YII 2

3.4.1. DESPLIEGUE DE YII 2 EN ENTORNO DE DESARROLLO Y PRODUCCIÓN

Yii 2 en el Template avanzado tiene pre configurado dos entornos: desarrollo y producción. Para configurar la aplicación en el entorno de desarrollo, se deberá de tener descargada una nueva aplicación con el Template avanzado y se ejecuta a través de la consola de comandos en la carpeta base del proyecto con el comando: ./init y se selecciona el modo development, como se muestra en Figura .

```
$ ./init
Yii Application Initialization Tool v1.0

Which environment do you want the application to be initialized in?

[0] Development
[1] Production

Your choice [0-1, or "q" to quit] :
```

Figura 18: Selección de entornos en Yii 2 para una aplicación con Template avanzado

Fuente: (Yii Suporters, 2015)

El entorno de desarrollo es creado con ficheros extras donde se pondrán las configuraciones personales del equipo de desarrollo y configuraciones del servidor que se manejan durante la fase de desarrollo de la aplicación. Son creados ficheros con el sufijo `-local` en su nombre y también ficheros `.gitignore`. Esto últimos permiten a Git la identificación de los archivos que contienen información sobre el entorno de desarrollo, descartándolos de las operaciones de subida en el repositorio en Internet. (Podlinov, 2014). Además, el estado `development` o desarrollo tiene todas las herramientas de desarrollo y de depuración activadas.

La Figura describe la estructura de entornos de una aplicación avanzada en Yii 2. Si el entorno de Yii 2 está en desarrollo, los ficheros con sufijo `-local` son creados y reescribirán las configuraciones similares que tengan sus homólogos.

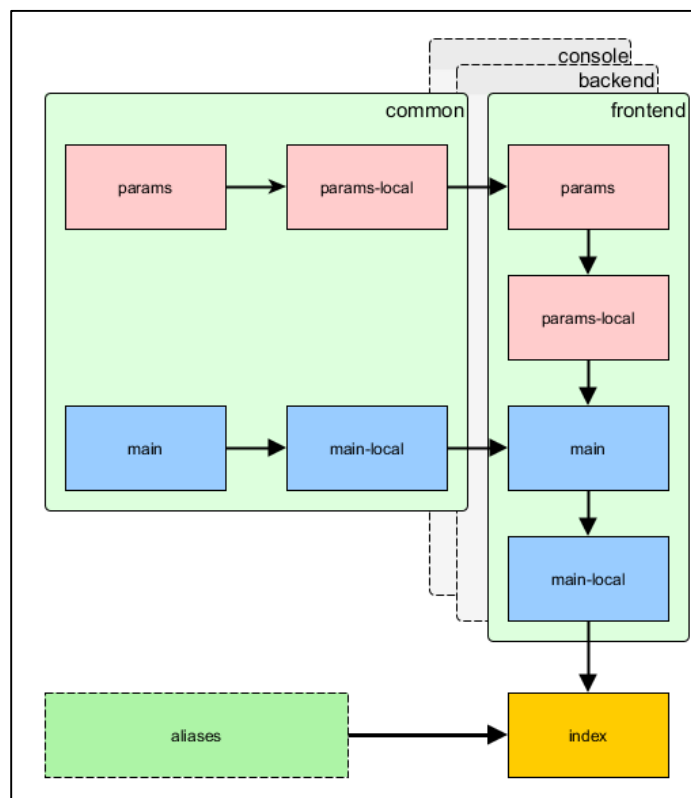


Figura 19: Estructura de entornos en Yii 2 con Template avanzado

Fuente: (Brandt, y otros, 2016)

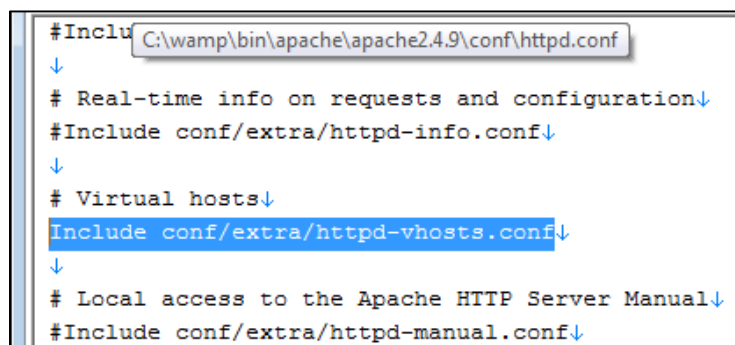
Según (Brandt, y otros, 2016) se creó esta estructura de entornos debido a que cada equipo de desarrollo tiene sus propias opciones de configuración, en desarrollo son usadas configuraciones, cuentas, contraseñas de acceso que no deberían de estar en un repositorio. Existen múltiples entornos en los servidores: desarrollo, prueba y producción. Cada uno de

estos tiene distintas configuraciones, las definiciones de estas opciones de configuración son repetitivas y se tardan mucho en realizar.

Para desplegar la aplicación en entorno de producción se debe de ejecutar el comando `./init` así como en la Figura y seleccionar `production`, este preguntará si se desea conservar los ficheros con el sufijo `-local` ya que posteriormente los eliminará. El estado `prod` o producción es para la implementación en producción y tiene apagadas las herramientas de desarrollo y depuración.

3.4.1.1. CONFIGURACIÓN DE ENTORNOS FRONTEND Y BACKEND EN LA FASE DE DESARROLLO

Al realizar las siguientes configuraciones se pretende que el Frontend y el Backend funcionen sobre un mismo dominio con el Template avanzado de Yii 2. Para ello, se debe de activar un host virtual el cual simulará el alojamiento de un proyecto en la Internet. Es necesario habilitar la línea de inclusión del archivo `httpd-vhosts.conf` en el fichero de configuración `httpd.conf` del servidor apache, así como lo indica la Figura 20.



```
#Include C:\wamp\bin\apache\apache2.4.9\conf\httpd.conf
↓
# Real-time info on requests and configuration↓
#Include conf/extra/httpd-info.conf↓
↓
# Virtual hosts↓
Include conf/extra/httpd-vhosts.conf↓
↓
# Local access to the Apache HTTP Server Manual↓
#Include conf/extra/httpd-manual.conf↓
```

Figura 20: Activación de host virtual

Fuente: Propio

El nuevo host virtual se debe configurar de acuerdo al directorio de publicación web sobre el cual se esté trabajando en el servidor Apache, la Figura da un ejemplo de cómo hacerlo con el fichero `httpd-vhost.conf`.

```

#   ErrorLog "logs/dummy-host2.example.com-error.log"
#   CustomLog "logs/dummy-host2.example.com-access.log" common
#</VirtualHost>↓
'
#advanced↓
<VirtualHost *:80>↓
    #nombre del host que agregamos en el archivo hosts↓
    ServerName advanced.loc ↓
    ServerAlias advanced.loc↓
    #ruta donde se guarda la pagina web↓
    DocumentRoot "C:/wamp/www/advanced" ↓
    <Directory "C:/wamp/www/advanced">↓
        #para leer el archivo .htaccess↓
        AllowOverride All↓
    </Directory>↓
</VirtualHost>↓

```

Figura 21: Configuración común para un host virtual

Fuente: Propio

Posterior a la configuración del virtual host se debe de indicar el nombre del dominio del proyecto relacionado con la IP local en el archivo hosts, esto permitirá la dirección del nombre de dominio hacia el proyecto alojado en el servidor local, así como lo muestra la Figura 22. Realizadas las configuraciones anteriores se procede a reiniciar el servidor.

```

# redirecting this is just a bad idea 127.0.0.1 localhost↓
# redirecting this is just a bad idea 127.0.0.1 localhost↓
# redirecting this is just a bad idea 127.0.0.1 localhost↓
127.0.0.1 localhost↓
127.0.0.1 advanced.loc ↓
↓

```

Figura 22: Configuración en el fichero hosts de Windows

Fuente: Propio

El control sobre el acceso a los ficheros y carpetas por parte del cliente debe de ser limitada. El framework dispone únicamente que sean de acceso el script de entrada y los recursos que se encuentran en la carpeta web en Frontend, Backend o en un Template básico (Yii Suporters, 2015).

Debido a que se usa el servidor Apache para las etapas de desarrollo y producción de la aplicación se deberá de configurar sobre los ficheros denominados .htaccess. Estos indican directrices específicas con las cuales deberá de comportarse el servidor cuando se encuentre con uno de estos ficheros (Foundation, 2016).

El Template avanzado deberá de tener tres ficheros .htaccess: uno en la raíz de la aplicación y otros dos sobre cada carpeta web del Frontend y Backend como lo indica la Figura 23 (Yii Software LLC, 2015).

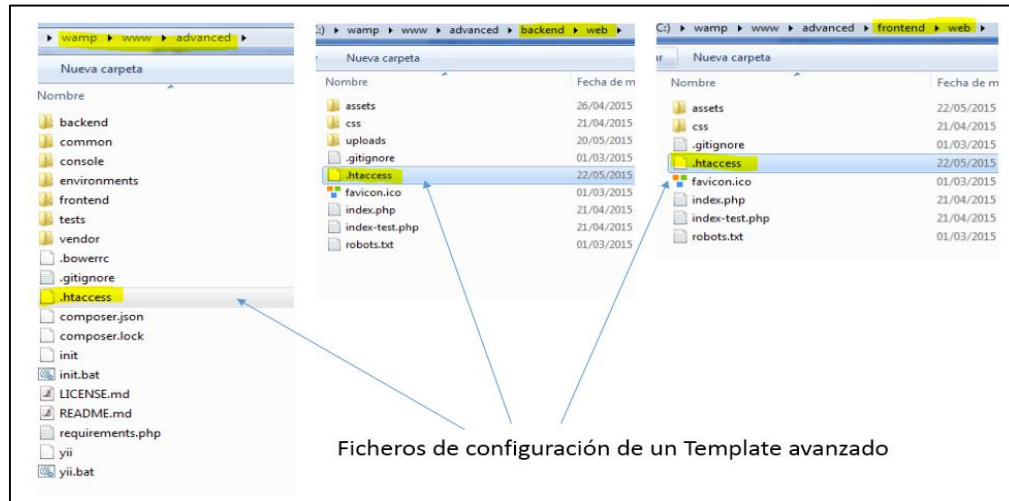


Figura 23: Ficheros de configuración del servidor en un Template avanzado Yii 2
Fuente: Propio

Además de proteger el acceso no deseado sobre archivos en el directorio del servidor, las reglas complementarias en los ficheros de las carpetas web proveerán de la configuración necesaria para activar la eliminación del nombre index.php sobre la url en el navegador.

El contenido del fichero .htaccess que se encuentra en la raíz del proyecto será similar al de la Figura 24.

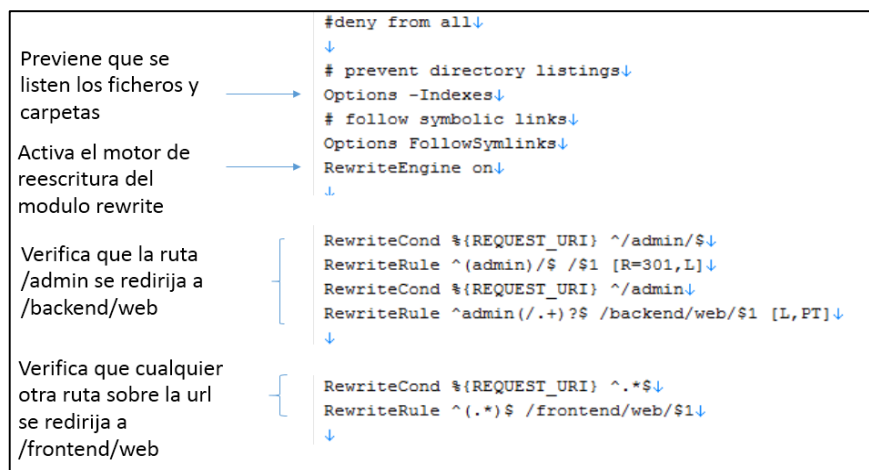


Figura 24: Fichero .htaccess que permite la redirección del Backend y Frontend en un mismo dominio

Fuente: (Yii-Suporters, 2016)

La Figura 24 anterior indica que el servidor permitirá reinterpretar las urls que se envíen desde el navegador. Primero, si en la ruta se indica la palabra admin/, se redireccionará a la carpeta backend/web en donde encontrará la página de inicio para este entorno y en el caso de indicar direcciones distintas a admin/, se redireccionará a la carpeta frontend/web.

La Figura 25 indica la configuración de los ficheros .htaccess que se encuentran en las carpetas web del Frontend y Backend.

```

# use mod_rewrite for pretty URL support
RewriteEngine on
# if a directory or a file exists, use the request directly
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
# otherwise forward the request to index.php
RewriteRule . index.php

```

Figura 25: Fichero .htaccess que permite la eliminación de index.php

Fuente: (Yii-Suporters, 2016)

Las nuevas configuraciones de re direccionamiento para el framework, están definidas en el archivo main.php de Backend donde se indica que la url base sobre la cual trabajará el sistema es “admin” como lo indica la Figura 26Figura .

```

'modules' => [
    'settings' => [
        'class' => 'backend\modules\settings\Settings',
    ],
],
'homeUrl' => '/admin',
'components' => [
    'authManager' => [
        'class' => 'yii\rbac\DbManager',
        'defaultRoles' => ['guest']
    ],
    'user' => [
        'identityClass' => 'common\models\User',
        'enableAutoLogin' => true,
    ],
],
'request' => [
    'baseUrl' => '/admin',
],
'urlManager' => [
    'enablePrettyUrl' => true, // mejora la visualizacion de simbolos
    'showScriptName' => false, // eliminara index.php de la url
],

```

Figura 26: Fichero main.php del Backend que indica las nuevas reglas de redirección

Fuente: Propio

Del mismo modo se tiene las configuraciones para el fichero main.php del Frontend.

```
return [
    'id' => 'app-frontend',
    'basePath' => dirname(__DIR__),
    'bootstrap' => ['log'],
    'controllerNamespace' => 'frontend\controllers',
    'homeUrl' => '/',
    'components' => [
        'user' => [
            'identityClass' => 'common\models\User',
            'enableAutoLogin' => true,
        ],
        'request' => [
            'baseUrl' => '',
        ],
        'urlManager' => [
            'enablePrettyUrl' => true, //limpia la url
            'showScriptName' => false, //desactiva visualizacion de index.php
        ],
    ],
];
```

Define como / la url para frontend

Indica que las peticiones entrantes deberán no deberán tener una url base

Activa el componente manejador de url

Figura 27: Fichero main.php del Frontend que indica las nuevas reglas de redirección

Fuente: Propio

En el servidor es necesario activar el módulo rewrite el cual se encuentra en el archivo de configuración de apache httpd.conf.

El resultado final será que se podrá acceder al entorno Frontend y Backend en un mismo dominio únicamente concatenando la palabra admin en la url del navegador, así como lo demuestra la Figura 28.

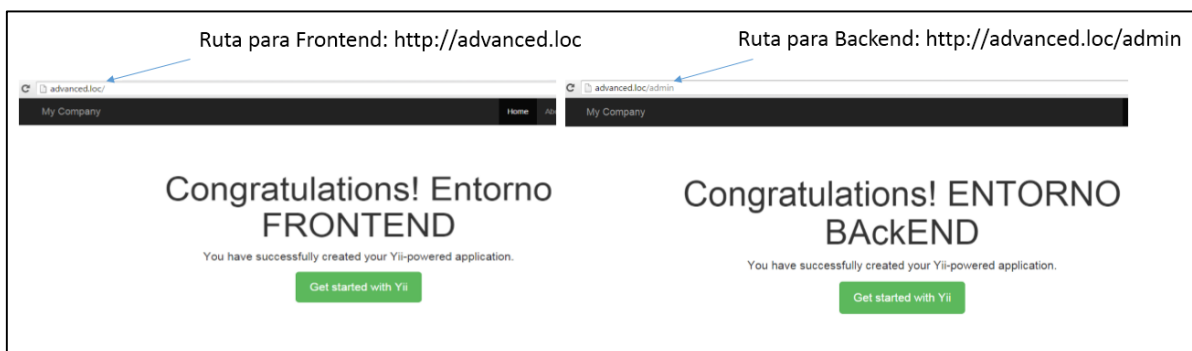


Figura 28: Backend y Frontend en un mismo dominio con un Template avanzado de Yii 2

Fuente: Propio

3.4.2. MANEJO DE WIDGETS DEL CORE DE YII 2

Al instalar los Templates básico o avanzado de Yii 2, automáticamente se descarga Bootstrap 3 y los widgets del framework. Con estos widgets la aplicación tendrá una interfaz moderna y llamativa. Los widgets permiten una integración más robusta con las características de Yii 2 y Bootstrap 3. Estos se los puede encontrar en `app/vendor/yiisoft/yii2/widgets` y en `app/vendor/yiisoft/yii2-bootstrap` como lo indica Figura 29.

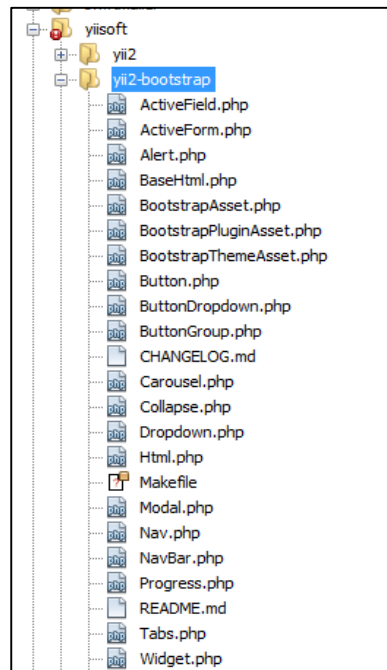


Figura 29: Ubicación de los widgets del core de Yii 2

Fuente: Propio

Todos los widgets son configurables a las necesidades del cliente, ya sea en forma o estilo para ello se necesita configurar las propiedades de estos. Las clases con las que se maneja cada componente de Bootstrap son agregadas automáticamente al momento de instanciar un Widget.

Dentro de los widgets más utilizados se tiene:

Widget ActiveForm

Este crea un formulario HTML es decir las etiquetas `<form>` y permite modificar la estructura de la grilla en donde se presentarán los nombres de los atributos, los errores y el componente de entrada de datos.

La Figura 30 muestra el código necesario para la visualización del Widget, este usa una propiedad denominada Template para la generación pragmática de cada atributo de un modelo o de un componente de entrada de datos.

```

$form = ActiveForm::begin([
    'layout' => 'horizontal',
    'fieldConfig' => [
        'template' => "{label}\n{beginWrapper}\n{input}\n{hint}\n{error}\n{endWrapper}",
        'horizontalCssClasses' => [
            'label' => 'col-sm-4',
            'offset' => 'col-sm-offset-4',
            'wrapper' => 'col-sm-8',
            'error' => 'col-sm-2',
            'hint' => 'col-sm-2',
        ],
    ],
]);

```

Figura 30: Configuración avanzada del widget ActiveForm de Yii 2

Fuente: (Yii-Suporters, 2016)

Para poder usar los campos de entrada como texto, textarea, radiobutton y demás que trae HTML 5 es necesario a través del widget ActiveForm representarlos con un atributo de un modelo, como lo indica la Figura 31. Este widget debe de ser envuelto por el ActiveForm para su correcto funcionamiento.

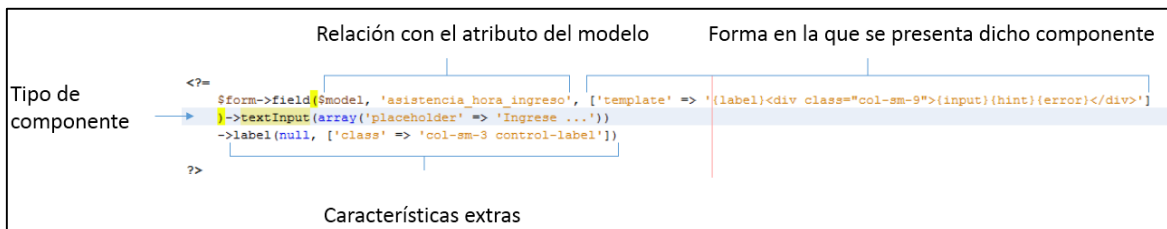


Figura 31: Widget ActiveForm de Yii 2

Fuente: Propio

Widget Alert

EL widget de alerta muestra notificaciones similares a las que tiene Bootstrap 3. Estas están unidas al componente de sesión del framework para poder así mostrar mensajes cada vez que el desarrollador lo vea necesario. La mejor forma de usar este widget es configurarlo en el fichero main.php de la carpeta views/layouts de tal modo que se muestren cada vez que se haga una acción de renderizado o refresco en la página, una vez configurado, se puede enviar alertas desde los controladores con el método Flash() del componente de sesión como lo indica la Figura 32.

```
Yii::$app->session->setFlash('error', 'There was an error sending email.');
```

Figura 32: Envío de alertas desde el controlador hacia la vista a través del objeto de sesión

Fuente: (Yii Suporters, 2015)

La Figura 33 siguiente muestra una forma de renderizar el widget de alertas. Esto es en el layout de cada entorno.

```
<?=  
Alert::widget(['options' => [  
    'style' => 'width:inherit !important; z-index:99; opacity:0.8; position: fixed;right: 10px;top: 100px;  
    'class' => 'alertMessage', 'duration' => 9000]])  
?>
```

Figura 33: Configuración del widget Alert

Fuente: Propio

Widget Modal

Este widget muestra una ventana modal según las especificaciones de Bootstrap 3. Es recomendable usar un solo widget de modal en el layout de la aplicación y cada vez que se lo requiera usar, se lo realizará a través de código javascript. La figura siguiente indica una forma de crear el widget, esta especifica las etiquetas que irán en el header y footer del mismo, así como identificadores y estilos para un fácil manejo con JQuery.

Inicio del widget MODAL	<code><?php Modal::begin([</code>	
Configuraciones de la cabecera del modal	<code> 'id' => 'mainModal', 'headerOptions' => ['id' => 'mainModalHeader', 'style' => 'text-align:center'</code>	Contenido HTML que se colocará en el header
Contenido HTML que se colocará en el footer	<code>], 'header' => '<h4 class="modal-title" id="labelMainModalHeader">LOADING... </h4>', 'footer' => '<button type="button" class="btn btn-sm btn-dark" data-dismiss="modal">Close</button>',</code>	
Opciones extras para el modal	<code> 'footerOptions' => ['id' => 'mainModalFooter',], 'size' => 'modal-lg', 'clientOptions' => ['backdrop' => 'static', 'keyboard' => FALSE] //para que se cierre unicamente con la x</code>	Opciones del pie del modal
Contenido del cuerpo del modal	<code>]); ?> <!--contenido del modal--> <center id="tagLoading"> <? = Html::img('@web/img/loading.gif') ?> </center> <!--fin contenido modal--></code>	
Fin del widget MODAL	<code><?php Modal::end(); ?></code>	

Figura 34: Configuración del widget de modal en Yii 2

Fuente: Propio

Para usar el Widget desde cualquier vista, se debe de configurar un archivo javascript que indique las operaciones requeridas con el modal del layout. De esta forma el código se hace más entendible y la aplicación más eficiente. La Figura 35 indica una función Javascript que coloca código HTML dentro del cuerpo del modal y a su vez muestra este en pantalla.

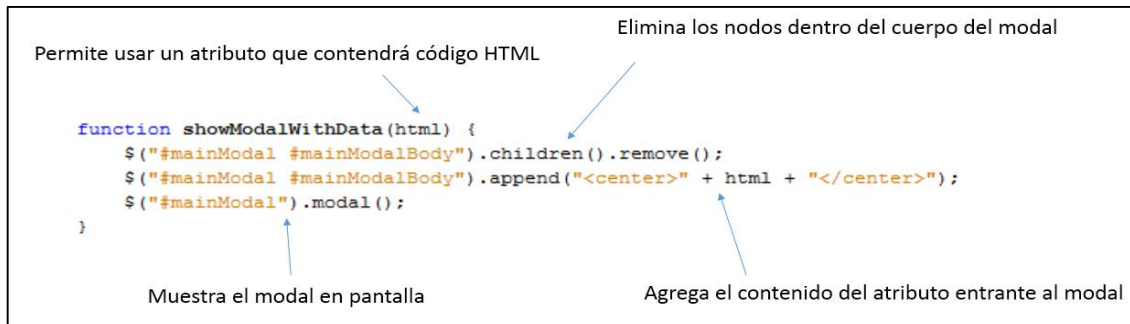


Figura 35: Función Javascript para llamar al widget de modal

Fuente: Propio

Widget Nav

Este widget permite mostrar un componente de navegación como la Figura 36. Se caracteriza en que cada submenú se lo representa como un array o bien se puede escribir directamente código HTML. Posee de opciones para habilitar la visibilidad, el label que representará al link y el link o url.

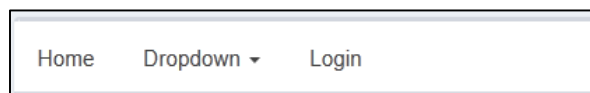


Figura 36: Ejemplo del widget de navegación

Fuente: (Yii-Suporters, 2016)

La Figura 37 indica las principales características de este widget.

```

echo Nav::widget([
    'items' => [
        [
            'label' => 'Home',
            'url' => ['site/index'],
            'linkOptions' => [...],
        ],
        [
            'label' => 'Dropdown',
            'items' => [
                ['label' => 'Level 1 - Dropdown A', 'url' => '#'],
                '<li class="divider"></li>',
                '<li class="dropdown-header">Dropdown Header</li>',
                ['label' => 'Level 1 - Dropdown B', 'url' => '#'],
            ],
        ],
        [
            'label' => 'Login',
            'url' => ['site/login'],
            'visible' => Yii::$app->user->isGuest
        ],
    ],
    'options' => ['class' => 'nav-pills'], // set this to nav-tab to get ta
]);

```

Key que indica un conjunto de menús

Array que representa a un menú

Etiqueta para el label

Url de redirección

Array que representa a un submenú

Característica de visibilidad

Figura 37: Configuración del widget Nav

Fuente: Propio

Widget NavBar

El NavBar permite colocar un label de identificación a la barra de navegación (Nav) como lo indica la Figura 38. Su funcionamiento es estético y se lo usa envolviendo al widget Nav.

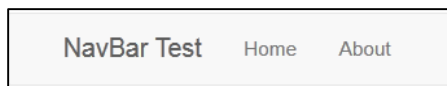


Figura 38: Ejemplo del widget NavBar

Fuente: (Yii-Suporters, 2016)

La Figura 39 muestra cómo debería de formarse ese widget.



Figura 39: Configuración del widget NavBar

Fuente: Propio

3.4.3. USO DE CACHE

El cache se entiende como el almacenamiento de: pequeños trozos de Strings, consultas a la bdd, objetos, llamadas a APIs o páginas renderizadas en alguna variable PHP, para posteriormente ser recuperadas y se representa como un objeto clave – valor.

El uso del cache promueve la aceleración de páginas dinámicas mediante la reducción de la carga de base de datos. Reutiliza las porciones de código generadas para reducir el tiempo de recuperación de los datos, ya que los procedimientos de creación de la misma se eliminan cuando ya se tiene dicho objeto creado en la cache (Yii Suporters, 2015).

Componentes de cache

Los componentes de cache son representaciones de los distintos tipos de almacenamiento en cache como: memoria, archivos y base de datos, estos son registrados como componentes de la aplicación para que sean configurables y accesibles globalmente. El acceso a estos componentes se lo realiza a través de: `Yii::$app->cache` y es posible usar más de un componente de cache en una misma aplicación (Yii Suporters, 2015). A continuación, se describen algunos tipos de cache que soporta Yii 2.

Tabla 5:
Tipos de cache que soporta Yii 2

Archivo	Origen	Uso
yii\caching\ApcCache	Usa la extensión de PHP APC	Eficiente y más rápida para usar en aplicaciones centralizadas. Ejm: un servidor no dedicado al balance de carga

Continua

yii\caching\DbCache	Usa una bdd para los registros, por defecto en SQLite3	Es configurable la bdd a elegir con su parámetro db
yii\caching\DummyCache		Se usa como una manera fácil de simplificar código necesario para comprobar la disponibilidad del uso de un cache. Se lo usa para sustituir temporalmente componentes de cache en específico en etapas de desarrollo cuando aún estos componentes no están bien definidos. Luego se lo puede cambiar con el componente necesario sin afectar el código
yii\caching\FileCache		Usa un fichero estándar para almacenar los datos. Es adecuado para almacenar grandes bloques de datos (páginas)
yii\caching\MemCache	Utiliza las extensiones de PHP memcache y memcached	La más rápida cuando la cache es manejada como una aplicación distribuida. Ejm: varios servidores con balance de carga
yii\redis\Cache	Basado en Redis, requiere la versión 2.6.12	Almacenamiento como mapa (clave – valor)
yii\caching\WinCache	Usa la extensión de PHP WinCache	Se la usa como un acelerador de PHP sobre servidores y sistemas operativos de Windows
yii\caching\XCache	Utiliza la extensión de PHP XCache	Para servidores en Linux sobre alta carga
yii\caching\ZendDataCache	Utiliza Zend Data Cache	Es un componente completo creado por Zend PHP. El api incluye: Almacenamiento, alimentación y eliminación de variables en cache, limpieza del cache, almacenamiento en disco o en memoria compartida, caching usando namespaces y configuración profunda de la carpeta de cache

Fuente: Propio

El acceso a los componentes de cache se lo realiza a través del api que Yii 2 dispone en su core. Todos los componentes de cache provienen de la clase padre yii\caching\cache.

Tabla 6:

Api para uso del cache en Yii 2

Clase \ método	Uso
yii\caching\Cache::get() \ get()	Recupera un elemento de datos de la memoria cache con una clave especificada, devuelve nulo cuando los datos no son encontrados, expirados o inválidos
yii\caching\Cache::set() \ set()	Almacena un elemento de datos identificado por una clave en la memoria cache
yii\caching\Cache::add() \ add()	Almacena un elemento en la cache, si la clave no se encuentra en cache
yii\caching\Cache::mget() \ mget()	Recupera un conjunto de datos según las claves especificadas
yii\caching\Cache::mset() \ mset()	Almacena un grupo de elementos en cache de acuerdo a las clave especificadas
yii\caching\Cache::madd() \ madd()	Almacena un grupo de elementos en la memoria, si la clave del elemento aún no existe, caso contrario se omite
yii\caching\Cache::exists() \ exists()	Devuelve un valor para indicar si la clave especificada se encuentra en memoria cache
yii\caching\Cache::delete() \ delete()	Elimina un elemento de datos indicando su clave
yii\caching\Cache::flush() \ flush()	Elimina todos los elementos de dato de la cache

Fuente: (Yii Suporters, 2015)

Cada elemento de cache se relaciona a partir de una clave la cual puede ser una cadena o un valor arbitrario. Para recuperar los valores en el cache se debe de llamar a esta clave.

Uso del cache en consultas de base de datos

Permite almacenar el resultado de consultas de base de datos, requieren de dos componentes: el primero es un componente de conexión de base de datos y el segundo es un componente de caché valido. El cacheo de datos es usado con DAO o a través de Active Record. La Figura muestra una forma de configurar el caching para consultas.

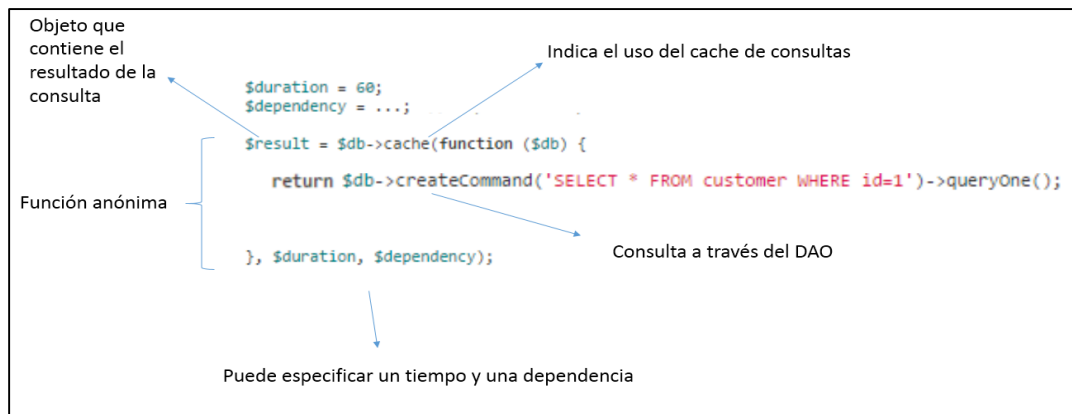


Figura 40: Uso del componente de cache para consultas de bases de datos

Fuente: (Yii-Suporters, 2016)

3.4.4. SEGURIDAD

3.4.4.1. FILTROS DE CONTROL DE ACCESO (ACF)

Este filtro de seguridad se implementa en la clase de core `yii\filters\AccessControl`, es utilizado en aplicaciones que necesitan algún control de acceso simple. Estos filtros pueden ser usados en controladores o módulos, mientras un usuario está enviando la petición de la ejecución de una acción, el ACF checará una lista de reglas de acceso para determinar si el usuario tiene permitido acceder a esa acción de petición (Yii-Suporters, 2016).

La Figura 41 muestra cómo se usa el ACF mediante el método `behaviors`, este método se lo debe de escribir en cada controlador del sistema.

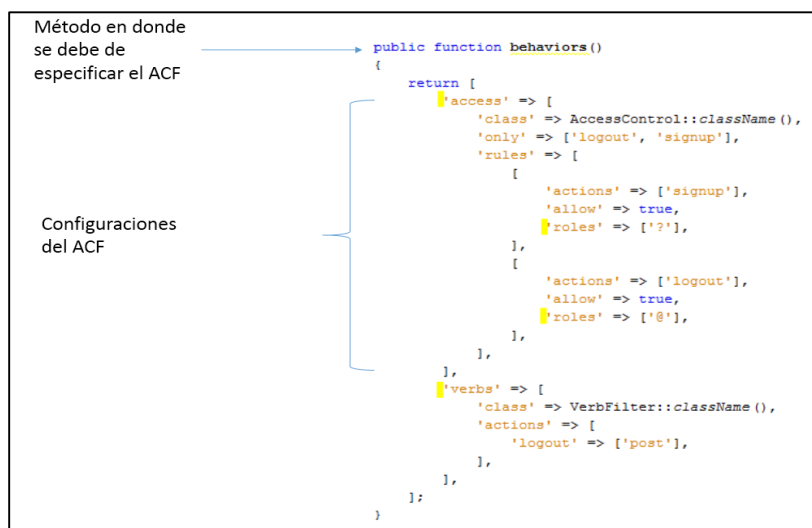


Figura 41: Configuraciones para el uso del ACF

Fuente: Propio

La opción `only` especifica que los ACF deberían únicamente ser aplicados a las acciones de: `login`, `logout` y `signup`. Las demás acciones de dicho controlador no están sujetas a este filtro.

La propiedad `rules` especifica características especiales para las acciones y tipos de usuarios.

En la Figura anterior se muestra que:

- La primera regla permite a todos los `guest` (invitados o usuarios aun no autenticados) acceder a las acciones `login` y `signup`. La opción `roles` contiene un símbolo o token de interrogación “?” el cual representa a los `guest user` o usuarios invitados.
- La segunda regla autentifica a los usuarios el acceso a la acción `logout`. El token “@” representa a los usuarios autenticados.

El valor `allow` indica si dicha acción es de permiso o denegación al usuario, si ninguna regla es marcada, significa que el usuario no es autorizado. Por defecto se toman las siguientes acciones:

- Si el usuario es `guest` (invitado), se llama a la acción de `yii\web\User::loginRequired()` para redirigir el navegador del usuario a la página de `login`.
- Si el usuario está autenticado, se lanzará una excepción `yii\web\ForbiddenHttpException`.

De igual manera el ACF puede monitorizar el acceso mediante la dirección IP, métodos `request` (`GET` o `POST`), modificar una acción de `callback matchCallback` y realizar acciones cuando se deniegue el acceso con `denyCallback` (Yii-Suporters, 2016).

3.4.4.2. USO DEL CONTROL DE ACCESO BASADO EN ROLES (RBAC)

El uso de RBAC del core de Yii conlleva a la realización de este en dos pasos. El primero es levantar los datos de autorización del RBAC y lo segundo es usar los datos de autorización para autorizar el chequeo de acceso en los lugares donde es necesario.

Se basa en una jerarquía en donde:

- Cada rol puede consistir de más roles y/o de una lista de acciones. Ejm: El rol Superadministrador tiene la permisología del rol Administrador, mas sus propios permisos.

- Los permisos son realizados de acuerdo a las acciones que se le permitan realizar a los roles y son especificados según urls pertenecientes a dichas acciones. Ejm: El rol de Gastrónomo debe de realizar las acciones concernientes a la preparación de recetas.
- Existen reglas que son clases específicas con procesos de lógica que permiten validar si dicho usuario podrá o no efectuar una acción. Estas se enlazan con un permiso a través de una dirección donde esté ubicada dicha clase.

El Figura 42 indica la jerarquía de permisos con la cual se debe de trabajar en el RBAC de Yii 2

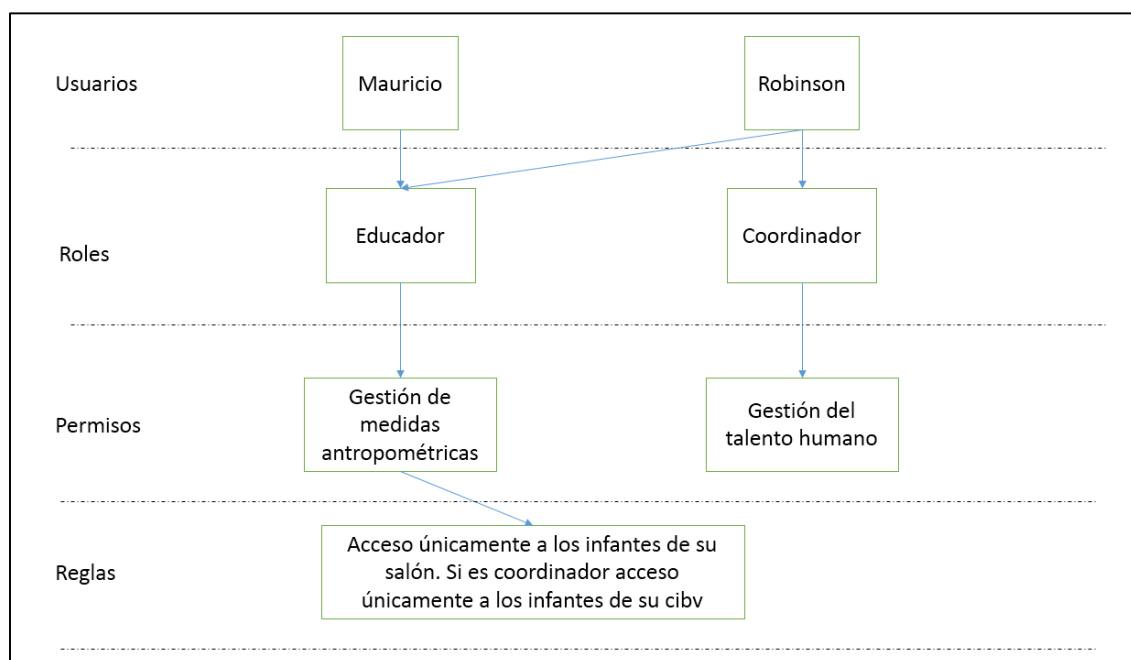


Figura 42: Diagrama de ejemplo de la permisología del RBAC en Yii 2

Fuente: Propio

La búsqueda en el árbol de permisos se la hace validando cada nivel desde las reglas hasta los roles, a través del método “can” que dispone el core de Yii 2.

Método para validar si un usuario puede realizar una acción, permiso o rol

```

public function can($permissionName, $params = [], $allowCaching = true)
{
    $auth = Yii::$app->getAuthManager();
    if ($allowCaching && empty($params) && isset($this->_access[$permissionName])) {
        return $this->_access[$permissionName];
    }
    $access = $auth->checkAccess($this->getId(), $permissionName, $params);
    if ($allowCaching && empty($params)) {
        $this->_access[$permissionName] = $access;
    }
    return $access;
}

```

Busca de forma recursiva en el árbol de permisos

Devuelve un boolean indicando la negación o aceptación del acceso

Figura 43: Método para validar los permisos del Rbac

Fuente: Propio

El método can() tiene que ser llamado en las acciones que deben de ser controladas

Acción a ser controlada

```

public function actionCreate() {
    if (Yii::$app->user->can('create-companie')) {
        $model = new Companies();
        -----
    }
}

```

Pregunta si el usuario que esta ejecutando esta acción puede ejecutar el permiso créate-companie

Figura 44: Configuración del método can() en las acciones del controlador

Fuente: Propio

Para que se pueda usar el RBAC, es necesario configurar el componente de aplicación authManager, el cual provee de dos formas de autorización: yii\rbac\PhpManager y yii\rbac\DbManager.

- PhpManager: usa un fichero PHP para almacenar los datos de autorización
- DbManager: usa una base de datos para almacenar los datos de autorización

Se consideraría usar PhpManager si no se tiene una administración dinámica de roles y permisos.

Para el componente DbManager es necesario tener las tablas que requiere el Rbac, para el S.O. Windows 7 estas se encuentran en `C:\wamp\www\advanced\vendor\yiisoft\yii2\rbac\migrations\`, o desde consola usando el comando de migración `yii migrate -- migrationPath=@yii/rbac/migrations` el cual creara automáticamente las tablas necesarias para el manejo del RBAC (Yii-Suporters, 2016)

Las tablas que requiere el Rbac son:

- `auth_item`: almacena los ítems de autorización: roles y acciones. Ejm: crear-compania, borrar-compania, actualizar-compania.
- `auth_assignment`: permite registrar los usuarios que tendrán permiso para realizar ciertos permisos. Ej.: El usuario de id 3 tiene permiso para la acción create-branch.
- `auth_item_child`: indica la relación padre-hijo de un permiso con otro permiso (rol permiso).
- `auth_rule`: almacena las reglas de acceso.

Es necesario la tabla user para que el Rbac del componente DbManager funcione.

3.4.4.3. CONFIGURACIÓN DEL MÓDULO DE ADMINISTRACIÓN DE RBAC

Debido a que Yii 2 no provee de una interfaz gráfica para la administración del Rbac incorporada en su core, se procede a configurar el módulo de <https://github.com/mdmssoft/yii2-admin>. Dicho módulo fue creado a partir de las clases de Rbac de yii2, por lo que se acopla perfectamente a su api (Munir, 2016). Después de instalar el módulo, es necesario tener activo el componente authManager en el archivo main.php de la consola o en common.php.

```
'components' => [  
    'authManager' => [  
        'class' => 'yii\rbac\DbManager', // or use 'yii\rbac\PhpManager'  
    ],  
],
```

Figura 45: Configuración del componente authManager

Fuente: (Munir, 2016)

Luego se procederá a indicar un nuevo módulo para el sistema, este se denominará admin y utilizará las clases de los archivos instalados.

```
'modules' => [  
  'admin' => [  
    'class' => 'mdm\admin\Module',  
    'layout' => 'left-menu', // it can be '@path/to/your/layout'.  
  ],  
],  
'as access' => [ //cuando el usuario no esta logueado  
  'class' => 'mdm\admin\components\AccessControl', //llama al ACF del RBAC  
  'allowActions' => [ //permite unicamente las acciones que se encuentren en  
    'site/*',  
  ]  
],  
],
```

Figura 46: Configuración del módulo para administración del Rbac

Fuente: (Munir, 2016)

Además, provee de un control de acceso el cual se describe como “as access”, la primordial función de este es redirigir a los usuarios invitados a páginas que se encuentren permitidas para ellos, y en caso de que quisiesen ingresar a otras direcciones que necesiten permisos de autenticación, se redirigirá a la página de login (Munir, 2016).

Si todo se configura bien se tendrá algo parecido a la siguiente Figura 47:

The screenshot shows a web interface for RBAC administration. On the left is a sidebar menu with options: Asignaciones, Roles, Permisos (highlighted), Rutas, Reglas, and Menús. The main content area is titled 'Permiso' and includes a 'Crear Permiso' button. Below the button, it says 'Mostrando 1-4 de 4 elementos.' and displays a table with the following data:

#	Nombre	Descripción	
1	actualizar_compania	Permite actualizar una compania	  
2	actualizar_compania_propia	Permite actualizar la compania unicamente al usuario que creo la compania	  
3	crear_compania	Permite crear una compania	  
4	permission_admin	Permisos para agregar, cambiar, modificar permisos, roles	  

Figura 47: Vista del módulo de administración del RBAC Yii 2-admin

Fuente: Propio

3.4.5. RESTFUL WEB SERVICES EN YII 2

Los servicios REST son un estilo de arquitectura de sistemas que permiten la comunicación bidireccional de datos entre distintas aplicaciones. Yii provee de un controlador específico para este tipo de requerimiento.

Usando el API RESTful de Yii 2, se implementa una API endpoint⁵⁵ en términos de una acción de un controlador y se usa un controlador para organizar las acciones que implementan los endpoint para un único tipo de recurso (Brandt, y otros, 2016).

La Figura 48 muestra un esquema sobre el cual trabajaría una (API) de Yii 2, esta se caracteriza en que la aplicación que se encuentra dispuesta en la red no dispone de vistas sino de un conjunto de endpoints que a través de una interface tipo REST, como lo es el controlador, se comunica con las otras aplicaciones mediante datos en formato xml o json.

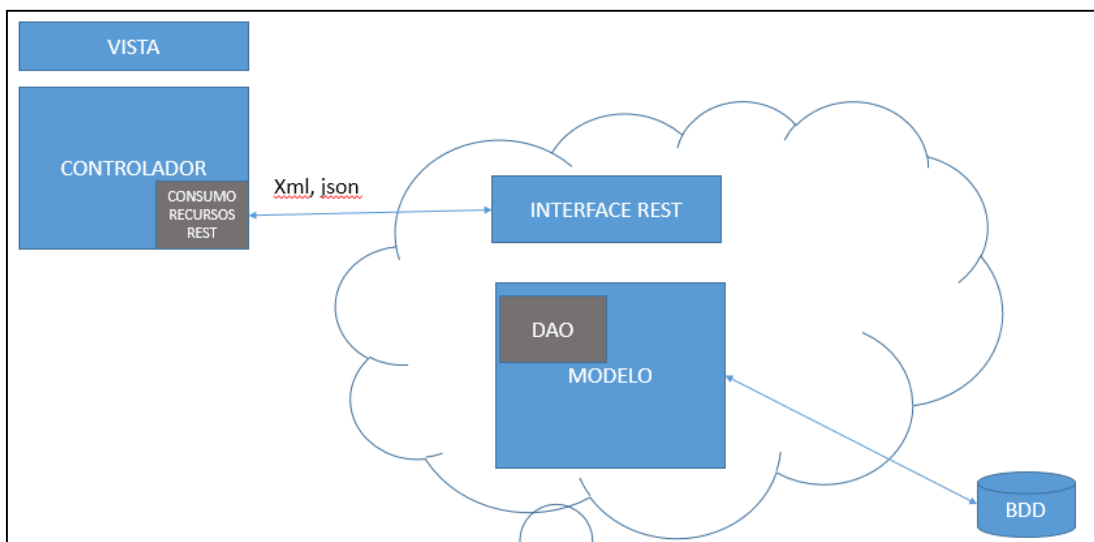


Figura 48: Esquema de aplicaciones que usan web services

Fuente: Propio

Los recursos de datos son representados como modelos, estos extienden desde `yii\base\Model` y es recomendable usar el Active Record. Dentro de las configuraciones del API se debe de usar la clase `yii\rest\UrlRule` para simplificar el ruteo de los API endpoints, es decir la forma en como aparecerán dispuestas a través de la url, esto se lo explica más a detalle en la Figura 49.

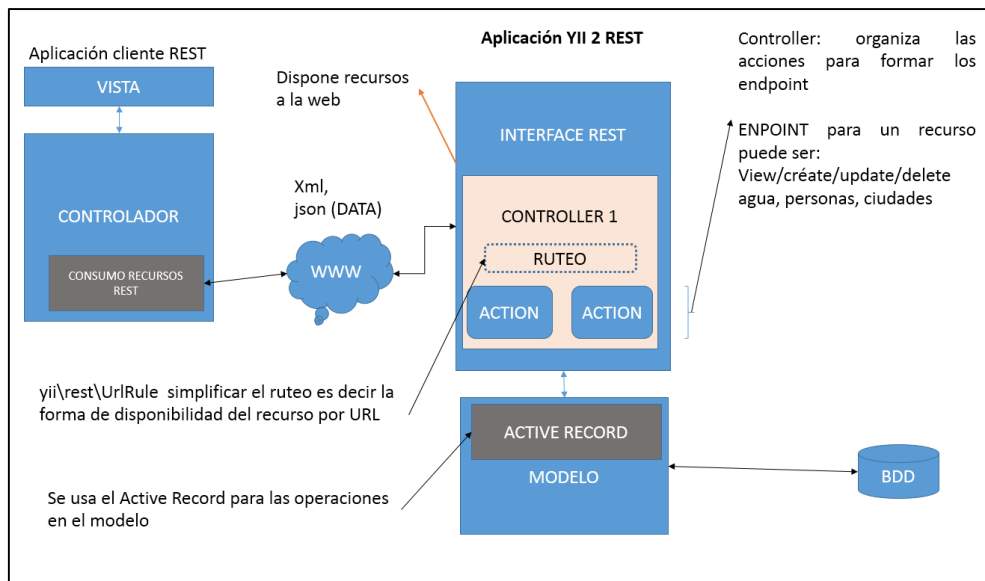


Figura 49: Esquema completo de comunicación entre un Frontend y un Api para Yii 2

Autor: Propio

3.4.5.1. CONFIGURACIÓN DEL ENTORNO PARA API YII 2

Las siguientes configuraciones se realizan sobre una aplicación con Template avanzado, esta aplicación deberá de tener una conexión con una base de datos. La tabla de base de datos de ejemplo será la de la Figura 50.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	p_a_id	int(11)			No	Ninguna	AUTO_INCREMENT
2	cliente_id	int(11)			No	Ninguna	
3	p_a_codigo_predio	varchar(45)	latin1_swedish_ci		No	Ninguna	
4	p_a_calles	varchar(45)	latin1_swedish_ci		No	Ninguna	
5	p_a_monto	varchar(45)	latin1_swedish_ci		No	Ninguna	
6	p_a_cancelado	enum('SI', 'NO')	latin1_swedish_ci		No	NO	

Figura 50: Base de datos de ejemplo para Api Yii 2

Fuente: Propio

Por seguridad se separa la lógica del API REST en otro directorio este puede ser una copia del frontend o backend que posteriormente se le denominara con el nombre de API. Según (Irawan, 2014) primero debemos de indicar a todo el sistema que existirá otro entorno que es el API y que va a estar situada en el directorio /api además, usará el nombre alias api, la

Figura 51 indica cómo se debe de configurar un nuevo archivo php llamado aliases.php el cual estará situado en las configuraciones comunes del sistema y tendrá el siguiente contenido.

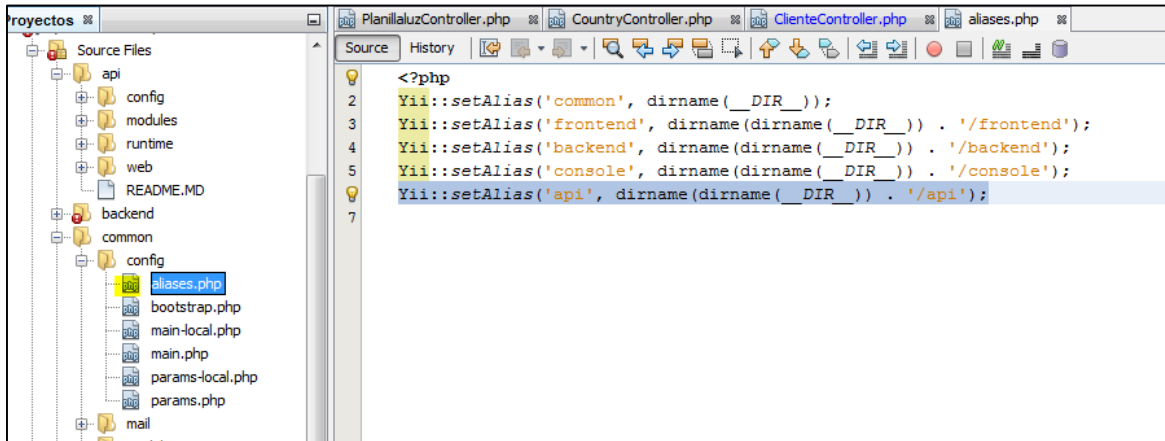


Figura 51: Configuración del entorno para API paso 1

Fuente: Propio

Se debe de agregar el archivo aliases.php al script de entrada en index.php de la carpeta web del entorno api.

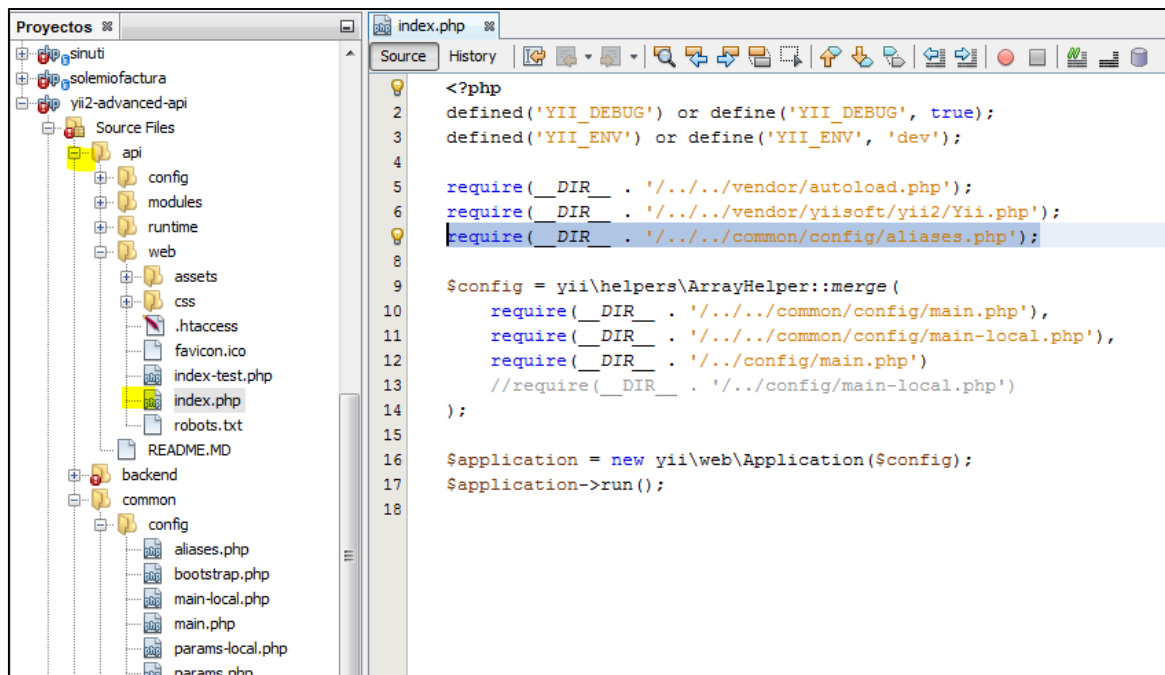


Figura 52: Configuración del entorno para API paso 2

Fuente: Propio

Se genera un nuevo módulo con GII y tendrá el aspecto final.

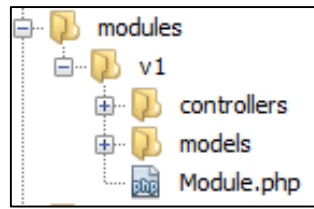


Figura 53: Estructura del módulo v1 para el entorno de API

Fuente: Propio

Dentro del módulo creado se procederá a crear controladores y modelos, no vistas ya que este no es un esquema de REST APPLICATION.

Cada controlador especificará un recurso, y cada action dentro de este especificará las acciones a realizarse, Yii trae una clase REST que permite tener una interfaz rápida para configuración de un API, ya que viene pre configurado operaciones de lectura escritura y borrado con los verbos get, post , etc como la Figura 54.

- **GET /countries:** list all countries
- **HEAD /countries:** show the overview information of country listing
- **POST /countries:** create a new country
- **GET /countries/AU:** return the details of the country AU
- **HEAD /countries/AU:** show the overview information of country AU
- **PATCH /countries/AU:** update the country AU
- **PUT /countries/AU:** update the country AU
- **DELETE /countries/AU:** delete the country AU
- **OPTIONS /countries:** show the supported verbs regarding endpoint /countries
- **OPTIONS /countries/AU:** show the supported verbs regarding endpoint /countries/AU.

Figura 54: Lista de ejemplo con las operaciones del API Rest de Yii 2

Author: (Irawan, 2014)

Se instancia una clase controlador que extienda de yii\rest\ActiveController. Esta deberá de tener un atributo publico \$modelClass el cual este apuntará a un modelo. Como lo indica la Figura 55.

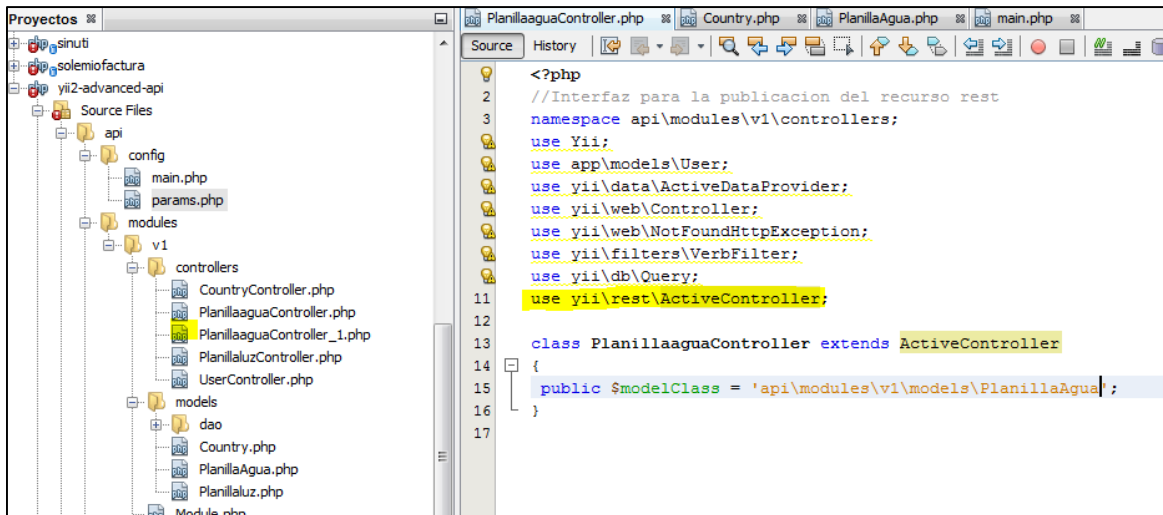


Figura 55: Configuración de interfaz web service tipo Rest en Yii 2

Fuente: Propio

Se debe de modificar el archivo de configuración principal del entorno api, para que pueda reconocer el módulo anteriormente generado, se comunique a través de objetos en JSON o XML y que disponga el servicio de los endpoints. Todo esto se lo visualiza en la Figura 56.

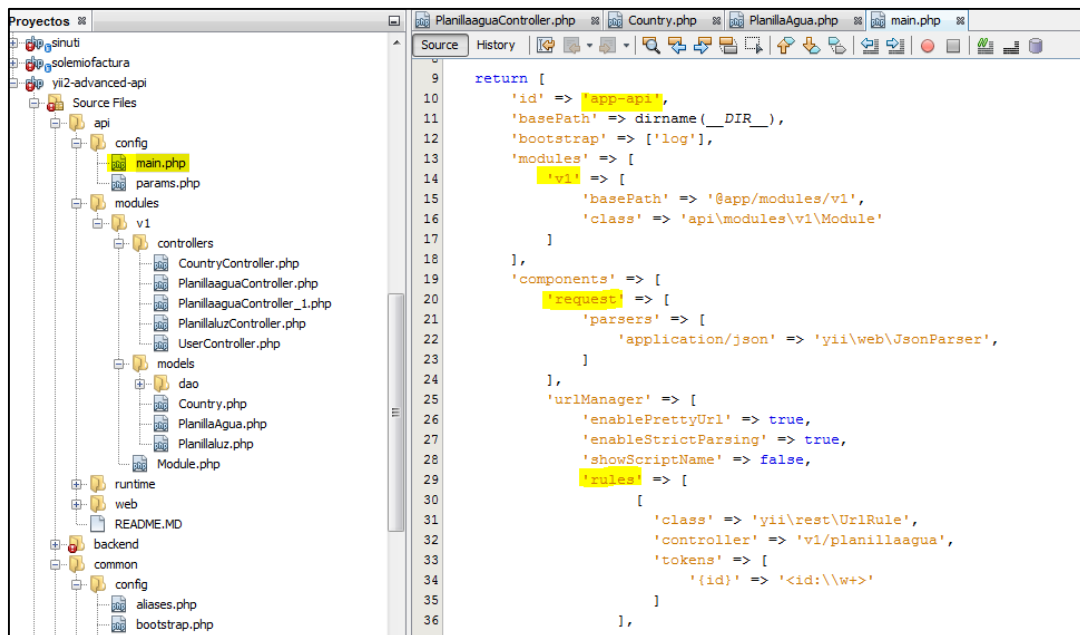


Figura 56: Configuración de config.php en el entorno Api para disponer de los servicios web

Fuente: Propio

3.4.5.2. USO DE YII2 PARA CLIENTE REST

Para un cliente Rest es necesario únicamente una aplicación básica o avanzada, pero con una instalación de extensión de yii2-httpclient a través de composer. Esta extensión permitirá el correcto manejo de las peticiones y respuestas. La Figura 57 muestra el paquete requerido.

```
"require": {
    "php": ">=5.4.0",
    "yiisoft/yii2": ">=2.0.5",
    "yiisoft/yii2-bootstrap": "*",
    "yiisoft/yii2-swiftmailer": "*",
    "yiisoft/yii2-httpclient": "~2.0.0"
},
```

Figura 57: Instalación del cliente Http para Yii 2

Author: (Klimov, 2016)

Según (Klimov, 2016) para consumir un servicio Rest es necesario que en cualquier controlador de la aplicación se genere una acción para realizar este procedimiento. Se debe de instanciar una clase Client() que es la que maneja las peticiones y proviene de la extensión instalada. Posterior a esto se procede a generar una nueva petición configurando el verbo, la url y formato de respuesta. Si el endpoint está disponible se generará una respuesta con el estatus 200 o OK, con los datos provenientes de esta respuesta se puede crear un objeto de modelo como un DataProvider que posteriormente se lo envía a la vista para ser mostrado en un widget. La Figura 58 muestra los pasos anteriormente indicados.

```
public function actionPlanillaagua()
{
    $client = new Client();
    //crea una nueva petición HTTP
    $response = $client->createRequest()
        ->setMethod('get') //configura el verbo de la operación
        //la URL del END POINT
        ->setUrl('http://localhost/yii2-advanced-api/api/web/v1/planillaagua')
    //envia la petición
    ->send();
    //la respuesta HTTP se gestiona una vez que se ejecuta send()
    //si la respuesta es exitosa, entonces prepara un dataprovider
    //para ser trabajado en u gridview en la vista
    if ($response->isOk) {
        $data = Json::decode($response->content);
        $dataProvider = new ArrayDataProvider([
            'allModels' => $data,
            'pagination' => [
                'pageSize' => 10,
            ],
        ],
    );
    }
    return $this->render('adminAgua', [
        'dataProvider' => $dataProvider,
    ]);
}
```

Figura 58: Configuración de un action para el consumo de un web service tipo Rest

Fuente: Propio

3.4.6. HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo fueron creadas para ayudar al programador en sus tareas de creación de ficheros, búsqueda de errores y generación de documentación como API (Yii Software LLC, 2015). Yii 2 provee de 3 herramientas:

Debug toolbar y debugger: es una herramienta y un componente que permiten tener de forma amigable información detallada sobre las operaciones que ejecuta el sistema, configuraciones del framework y de PHP, visualiza el contenido de las peticiones y respuestas HTTP, la ejecución de las consultas en la bdd y gestiona los logs que genera la aplicación.

Generador de código gii: permite generar código de forma genérica de módulos, controladores, vistas, modelos y extensiones para Yii 2.

Generador de documentación tipo API: es una extensión que provee de un generador de documentación en forma de API.

CAPÍTULO IV

4. IMPLEMENTACIÓN DE SISTEMA DE SEGUIMIENTO NUTRIMENTAL INFANTIL FASE DOS

4.1. FASE DE PLANIFICACIÓN

4.1.1. HISTORIAS DE USUARIO

Las historias de usuario son pequeñas descripciones del comportamiento del sistema desde el punto de vista de un usuario del sistema, sirven para la comunicación entre los clientes y los desarrolladores ya que a partir de estos se determinan primero las iteraciones con prioridad más alta (Shore & Warden, 2007).

4.1.1.1. HISTORIAS DE USUARIO GASTRÓNOMO

Se presenta a continuación la historia de usuario número uno para el usuario con rol gastronómico. Esta permitirá obtener las porciones alimenticias.

Tabla 7:
Historia de usuario para la gestión de recetas

HISTORIA DE USUARIO			
1. Numero de historia:	1	2. Usuario:	Lic. Juan Felipe Bixby
3. Nombre de la historia:	Gestión de recetas (Módulo Gastronomía)	4. Riesgo	Exceso de información para procesar sobre las porciones infantiles para cada ingrediente.
5. Prioridad (alta, media, baja)	Media	6. Punto Estimado:	
7. Iteración Asignada:	1	8. Fecha:	23/11/2015
9. Programador Responsable:	Mauricio Chamorro		
10. Descripción			
El Lic. Bixby procedió a indicar la diferencia entre recetas y menús, indicando que para las recetas se los debería de realizar a través de una lista de ingredientes con medidas en gramos, sean estos líquidos, semi-sólidos o sólidos. Se indicó que el procedimiento para la generación de recetas debería de ser similar como la generación de menús en cuanto a su presentación.			
11. Observaciones			
Se entregó un documento en Excel para que el Lic. Bixby entregase porciones cuantificadas en gramos y medidas caceras para infantes de todos los alimentos con los cuales el sistema trabaja.			

Fuente: Propio

La presente historia de usuario muestra una categorización de las cantidades de los platillos clasificados de acuerdo a su tipo acorde a lo que el menú semanal requerido por el MIES en su planificación semanal.

Tabla 8:
Historia de usuario que indica la categorización de porciones

HISTORIA DE USUARIO				
1. Numero de historia:	2	2. Usuario:	Lic. Juan Felipe Bixby / Gastrónomo	
3. Nombre de la historia:	Nuevo sistema de categorización para las porciones infantiles (Módulo Gastronomía)	4. Riesgo	Se deja a criterio del personal técnico las cantidades de porciones infantiles ya que no pueden estar acordes a lo que el Mies indica.	
5. Prioridad (alta, media, baja)	Alta	6. Punto Estimado:		
7. Iteración Asignada:	1	8. Fecha:	24/11/2015	
9. Programador Responsable:	Mauricio Chamorro			
10. Descripción				
Debido al extenso trabajo que presentó la obtención de porciones en gramos para cada ingrediente o platillo de la tabla de composiciones de alimentos del Ecuador. El Lic. Bixby indicó que se debe de categorizar por platos, así como se lo realiza en catering.				
De lo que se obtuvo los siguientes datos:				
Grupo	Nombre o receta tipo	Ejemplos	Cantidad máxima sugerida [gr]	Equivalente Categorización Mies
Entradas	En Ecuador no se tiene como cultura el servir una entrada (mariscos, sushi), por lo que la entrada es la SOPA			
Sopas	cremas	crema de zanahoria, crema de champiñones	100	a.Sopa
	caldos	consomé	100	a.Sopa
	sopas	sopa de quinua, sopa de arroz de cebada	100	a.Sopa
Platos fuertes	carnes [Proteínas]	estofados, aves y mariscos	50	b.2 Carne o huevo o leguminosa
	guarniciones	arroz, papas, fideos, mote, verde, yuca	50	b.1 Cereal o tubérculo
	ensaladas y menestras (frías)	lechuga con tomate y cebolla, encurtido	30	b.3 Ensalada o guiso
	ensaladas y menestras (calientes)	guiso de verduras	30	b.3 Ensalada o guiso
Breaks	alimentos de media mañana / media tarde	ensalada de frutas	40 sólido, 40 líquido total 80	c. Fruta o jugo natural
		bolón, majado	80	b (preparación sólida)
		colada de plátano	80	a (preparación líquida o espesa)
	coladas, batidos, avenas	manzana, pera, uvas	100	c. Fruta o jugo natural

Frutas o jugos	fruta	jugo de naranjilla	100	c. Fruta o jugo natural
	jugos	pan, galletas	40	b (preparación solida)
Otros	postres			

Además sugirió que el sistema debería de seguir los siguientes pasos para la creación de platillos:

1. Elegir el grupo
2. Elegir la receta o nombre
3. Elegir los ingredientes
4. Verificar los valores nutricionales del platillo conforme se ingresen los ingredientes con su respectivo gramaje
5. Elegir la cantidad de comensales
6. Obtener un aproximado de la cantidad total de ingredientes para la preparación del platillo para ese grupo de comensales

11. Observaciones

En el caso de que los infantes requieran mayor cantidad de nutrientes en sus dietas debido a enfermedades, la preparación del platillo debe de ser realizada a criterio del usuario [especialista] quien deberá tener los conocimientos necesarios para estos casos.

También, el Lic. Bixby indicó que este procedimiento de preparación de recetas debe de realizarse por un usuario con las capacidades necesarias para dar su criterio técnico al momento de la preparación del mismo.

Fuente: Propio

4.1.1.2. HISTORIA DE USUARIO RBAC

Esta historia de usuario indica la fuente para crear los casos de uso referentes a los permisos y roles del sistema.

Tabla 9:

Historia de usuario para el Rbac

HISTORIA DE USUARIO			
1. Numero de historia:	3	2. Usuario:	Dra. Rocío Olivo / Coordinador Gad
3. Nombre de la historia:	Planificación Módulo RBAC	4. Riesgo	Información desactualizada o que no esté acorde a los procesos en la vida real
5. Prioridad (alta, media, baja)	Alta	6. Punto Estimado:	
7. Iteración Asignada:	2	8. Fecha:	21/12/2015
9. Programador Responsable:	Mauricio Chamorro		

Continúa 

10. Descripción
La Doctora requiere que cada usuario descrito en el sistema realice las tareas a las cuales está sujeto de acuerdo a los orgánicos funcionales como por ejemplo: Normativa de la externalización de alimentación de los Cibv, así como de las normas técnicas que dicta el MIES para el desarrollo infantil integral.
11. Observaciones
No provee de información física o digital, se indicó que toda la información concerniente para levantar el RBAC se encuentra en la documentación que las instituciones tienen publicadas en internet.

Fuente: Propio

4.1.1.3. HISTORIAS DE USUARIO NUTRICIONISTA

La historia de usuario número cinco aporta con requerimientos sobre el incremento del módulo de recetas y bitácora.

Tabla 10:

Historia de usuario del módulo bitácora de consumo

HISTORIA DE USUARIO			
1. Numero de historia:	5	2. Usuario:	Dra. Susana Castillo Lara / Nutricionista
3. Nombre de la historia:	Planificación módulo bitácora	4. Riesgo	No se tenga el personal técnico en el área de gastronomía
5. Prioridad (alta, media, baja)	Alta	6. Punto Estimado :	
7. Iteración Asignada:		8. Fecha:	26/01/2015
9. Programador Responsable:	Mauricio Chamorro		
10. Descripción			
La Dra. Susana Castillo Lara ve como una forma de mejorar el sistema el incremento de módulos para la preparación de comidas o recetas. También necesita que se obtenga de alguna forma el porcentaje de adecuación del consumo ideal y el poder estandarizar las porciones de los niños.			
11. Observaciones			
No existen observaciones.			

Fuente: Propio

Esta historia de usuario indica cual es el procedimiento para obtener el porcentaje de adecuación para el desarrollo del módulo de bitácora de consumo infantil.

Tabla 11:

Historia de usuario que indica el procedimiento del porcentaje de adecuación

HISTORIA DE USUARIO						
1. Numero de historia:	6			2. Usuario:	Dra. Susana Castillo Lara / Nutricionista	
3. Nombre de la historia:	Procedimiento para porcentaje de adecuación (Módulo Bitácora)			4. Riesgo	Exceso de información en una misma vista	
5. Prioridad (alta, media, baja)	Alta			6. Punto Estimado :		
7. Iteración Asignada:	3			8. Fecha:	27/01/2016	
9. Programador Responsable:	Mauricio Chamorro					
10. Descripción						
La Dra. Susana Castillo Lara indica que para obtener este parámetro es necesario realizar un cálculo entre lo que consume el infante sobre el adecuado para su edad (macro y micronutrientes) y esto obtener como porcentaje. Este procedimiento se lo debe de realizar por cada día de alimentación en los Cibv. Este parámetro servirá para realizar sugerencias de alimentación y ver el desequilibrio nutricional. El formato sugerido para esta presentación es la siguiente:						
		Lunes	Martes	Miércoles	Jueves	Viernes
Consumido						
Ideal						
% Adecuación						
11. Observaciones						
No existen observaciones.						

Fuente: Propio

4.1.2. DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

Documento de especificación de requisitos de software (ERS) es un estándar que promueve la sociedad de computación de la IEEE sirve como la base para la estimación de costes y planificación, de punto de referencia para procesos de verificación y validación y para ayudar a los desarrolladores a entender que quiere exactamente el cliente (Vidal, 2016).

En el anexo C se describe el documento ERS generado para el desarrollo de este estudio.

4.2. FASE DE DISEÑO

4.2.1. CASOS DE USO

El caso de uso es un tipo de diagrama de UML⁵⁶ que permite indicar como el usuario o actor interactúa con el sistema en desarrollo (Caro, 2016). Se presenta a continuación los diagramas creados a través de los requisitos del sistema.

4.2.1.1. CASO DE USO GESTIÓN MÓDULO GASTRONOMÍA

La Figura 59 muestra las tareas que el usuario con rol Gastrónomo puede realizar en el sistema.

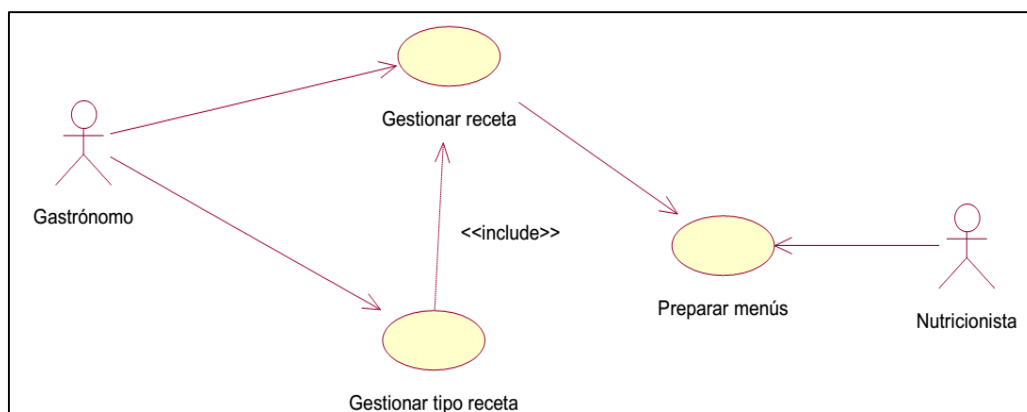


Figura 59: Caso de uso gestión módulo gastronomía

Fuente: Propio

La Tabla 12 describe el caso de uso de gestión del módulo gastronomía.

Tabla 12:

Descripción caso de uso gestión módulo gastronomía

CASO DE USO:	Gestión módulo gastronomía
Descripción	El Gastrónomo es el encargado de la preparación de recetas, puede editar las recetas, eliminar ingredientes o eliminar recetas. También puede crear, eliminar, actualizar y eliminar los tipos de recetas. Una vez que se crea la receta se lo puede usar para la creación de menús.
Actor	Gastrónomo

Continua

Condiciones previas	<ul style="list-style-type: none"> • Estar registrado y tener rol gastrónomo en Sinuti.
Flujo básico eventos	<ul style="list-style-type: none"> • Registrar, actualizar o eliminar recetas. • Registrar, actualizar o eliminar tipos de receta.
Flujos alternativos	<p>Se mostrará un mensaje error si: intenta eliminar un tipo receta que ya esté en uso en una receta o intenta eliminar una receta que ya sea usado para la preparación de un menú.</p> <p>No se puede visualizar recetas sin el registro previo de alimentos o ingredientes en el sistema.</p>
Escenario clave	<p>Tener disponibles los ingredientes o alimentos para la preparación de recetas.</p> <p>Tener disponibles los tipos de recetas.</p>
Post-condiciones	Dispone de nuevas preparaciones para la preparación de cartas.

Fuente: Propio

4.2.1.2. CASO DE USO GESTIÓN MÓDULO RBAC

En la Figura 60 se puede observar las tareas que puede realizar en el sistema el usuario con privilegios de Superadministrador.

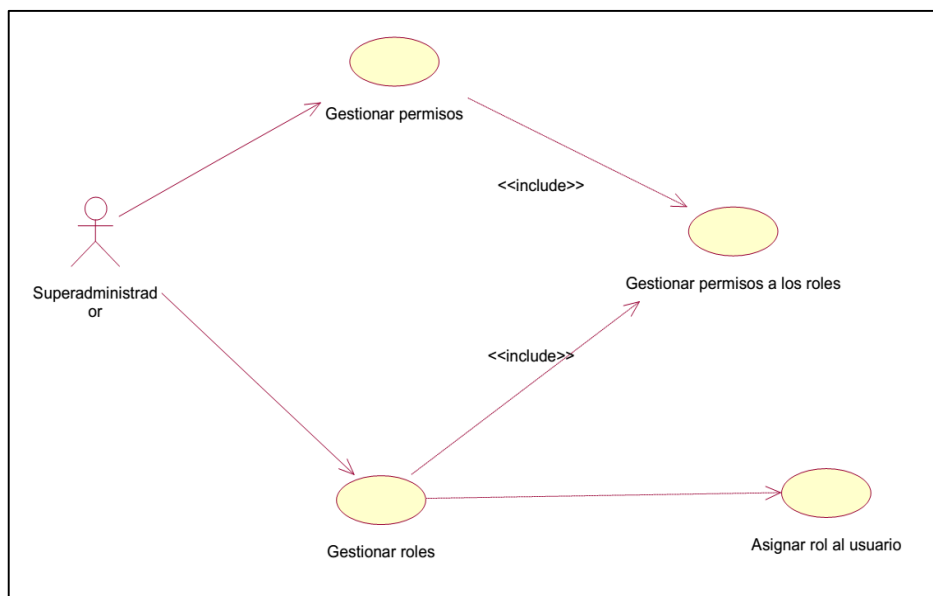


Figura 60: Gestión módulo Rbac

Fuente: Propio

La Tabla 13 Describe el caso de uso de gestión del módulo de Rbac

Tabla 13:

Descripción caso de uso gestión Rbac

CASO DE USO:	Gestión módulo Rbac
Descripción	El Superadministrador es el encargado de crear, actualizar y borrar permisos, roles del sistema y permisos a los roles. Este también puede de asignar el rol al usuario.
Actor	Superadministrador.
Condiciones previas	<ul style="list-style-type: none"> • Estar registrado y tener rol Superadministrador en Sinuti.
Flujo básico eventos	<ul style="list-style-type: none"> • Registrar, actualizar o eliminar permisos • Registrar, actualizar o eliminar roles. • Registrar, actualizar o eliminar permisos a los roles. • Asignar y eliminar a usuarios de los roles
Flujos alternativos	Ninguno
Escenario clave	Tener permisos disponibles para asignarlos a un rol. Tener roles disponibles para asignarlos a un usuario.
Post-condiciones	Ninguno

Fuente: Propio

4.2.1.3. CASO DE USO GESTIÓN MÓDULO BITÁCORA

En la Figura 61 se puede observar las tareas que el usuario con rol Educador realiza en el sistema.

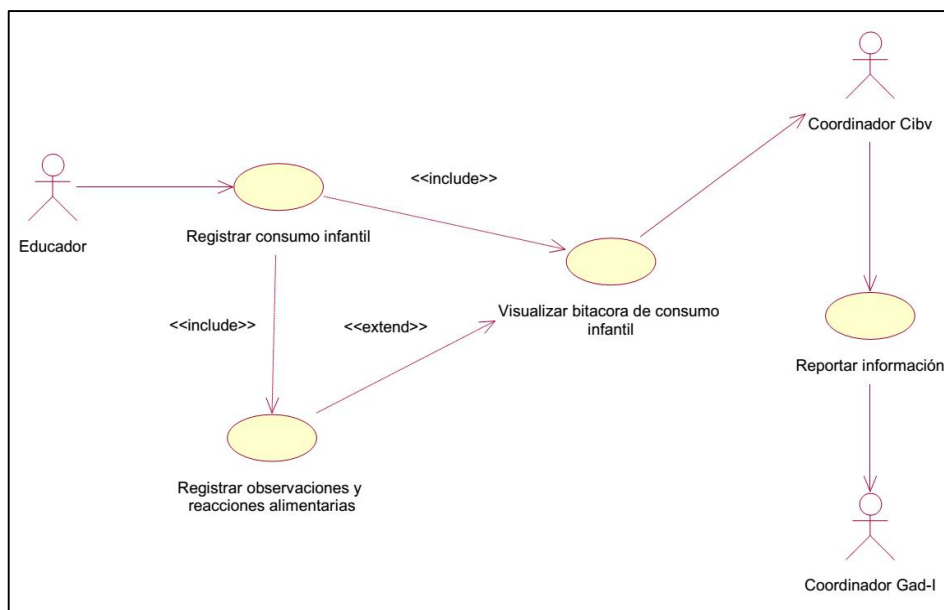


Figura 61: Gestión módulo Bitácora

Fuente: Propio

La Tabla 14 describe el caso de uso de gestión del módulo bitácora.

Tabla 14:

Descripción caso de uso gestión bitácora

CASO DE USO:	Gestión módulo bitácora
Descripción	El educador registra y actualiza los datos del consumo infantil, así como de las observaciones y reacciones alimentarias, con estos datos se puede visualizar la bitácora de consumo infantil.
Actor	Educador
Condiciones previas	<ul style="list-style-type: none"> • Estar registrado y tener rol Educador en Sinuti. • Disponer de infantes registrados en el Cibv del Educador. • Disponer de una carta o menú semanal asignado para el consumo de infantil.
Flujo básico eventos	<ul style="list-style-type: none"> • Registrar y actualizar el consumo infantil de su sector asignado en el periodo actual. • Registrar y actualizar observaciones y reacciones alimentarias de su sector asignado. • Visualizar bitácora de consumo de su sector asignado.
Flujos alternativos	Ninguno
Escenario clave	Tener infantes disponibles en su sector asignado. Tener asignado un menú semanal al Cibv.
Post-condiciones	Ninguno

Fuente: Propio

4.2.2. DIAGRAMAS DE PROCESO

El diagrama de procesos indica las secuencias de pasos que tendrán que realizar los usuarios para cumplir un objetivo, muestra el flujo de la información y como esta se produce. A continuación se presentan los diagramas generados para el desarrollo de Sinuti Fase 2.

4.2.2.1. DIAGRAMA DE PROCESO GESTIÓN DE RECETAS

La Figura 62 muestra los pasos que debe realizar el usuario con rol Gastrónomo para gestionar las recetas.

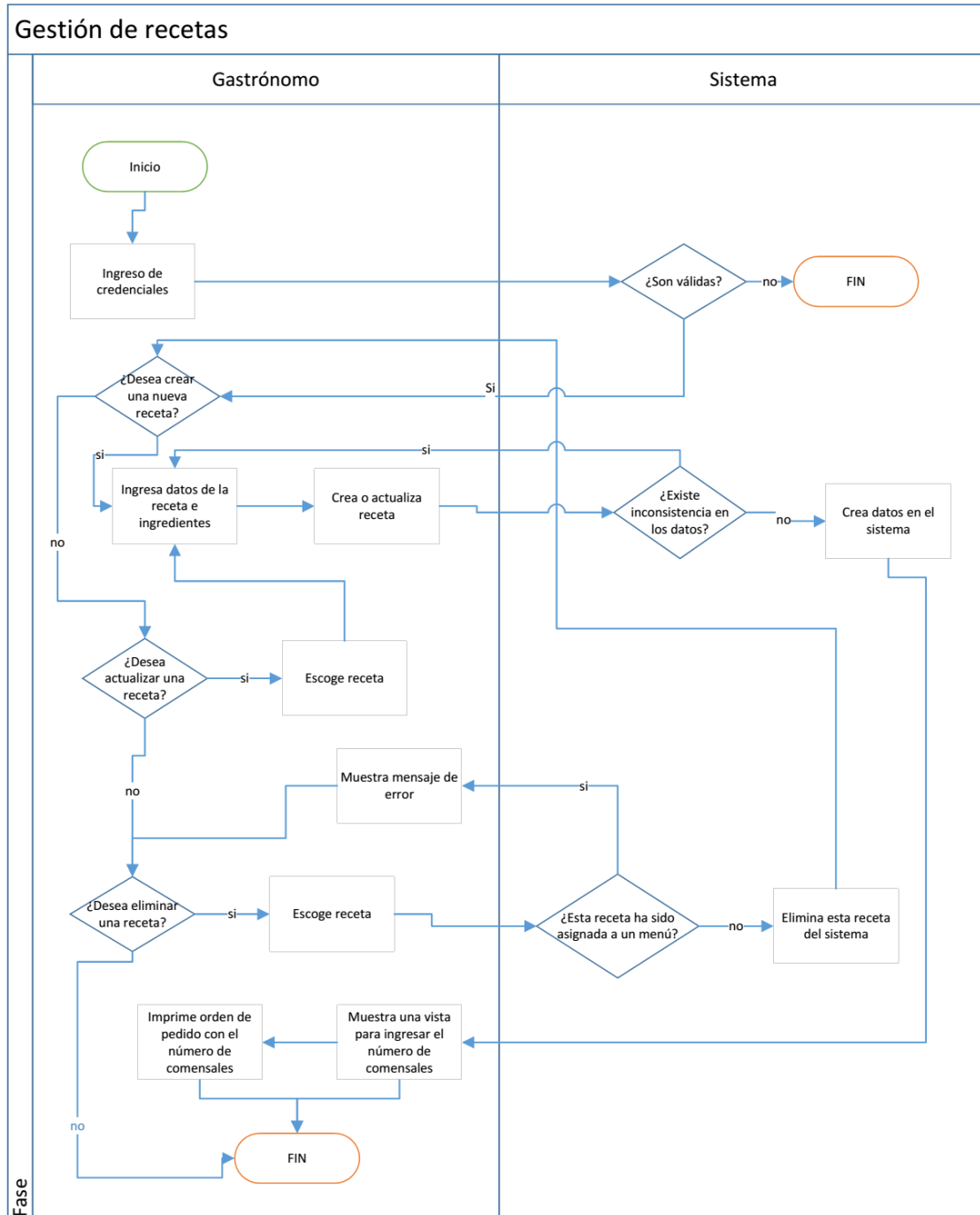


Figura 62: Diagrama de proceso gestión recetas

Fuente: Propio

4.2.2.2. DIAGRAMA DE PROCESO GESTION DE TIPO DE RECETA

La Figura 63 muestra los pasos que debe realizar el usuario con rol Gastrónomo para gestionar los tipos de receta.

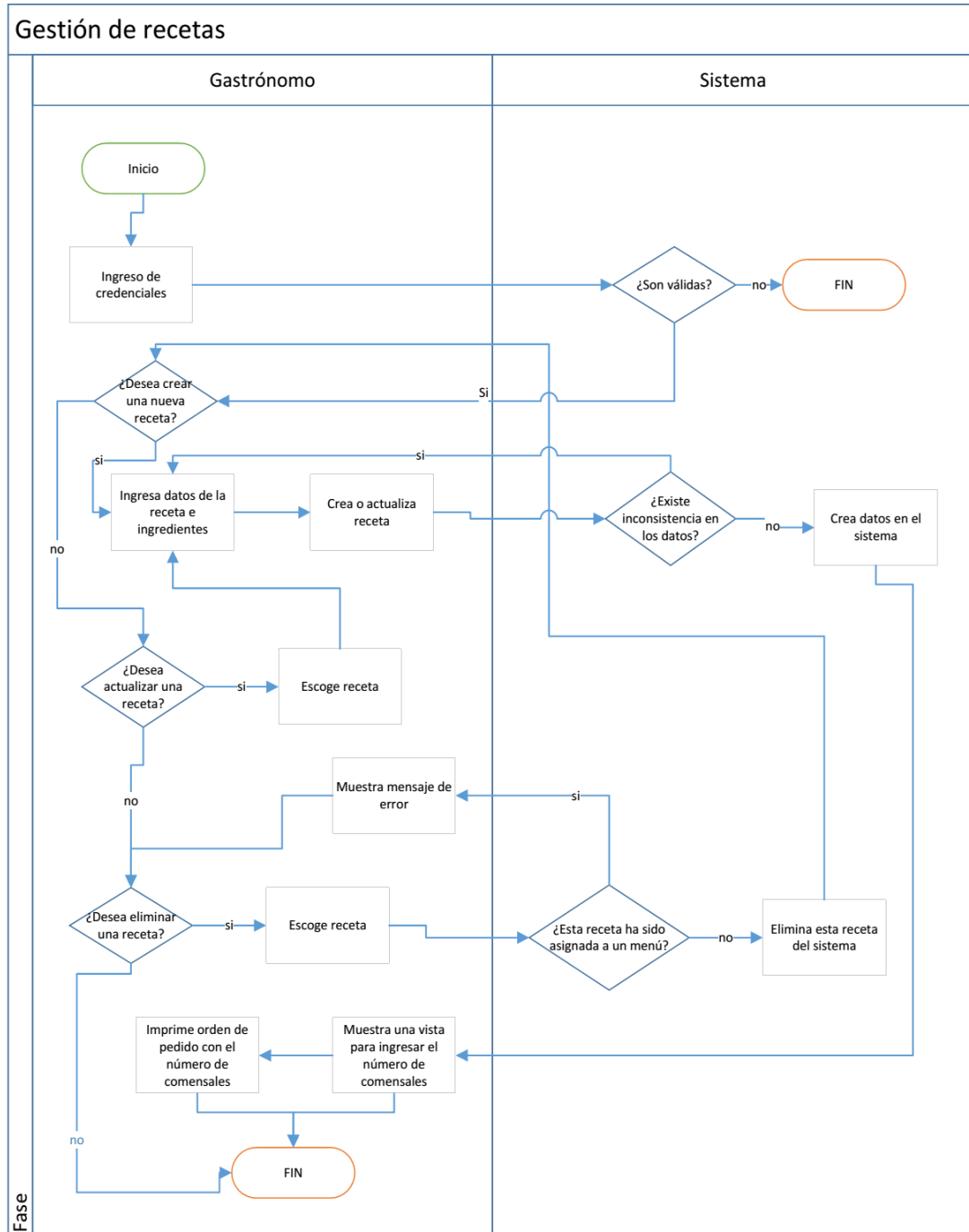


Figura 63: Diagrama de proceso gestión tipo receta

Fuente: Propio

4.2.2.3. DIAGRAMA DE PROCESOS GESTION DE PERMISOS PARA ROLES

La Figura 64 muestra los pasos que debe realizar el usuario con rol Superadministrador para gestionar los roles.

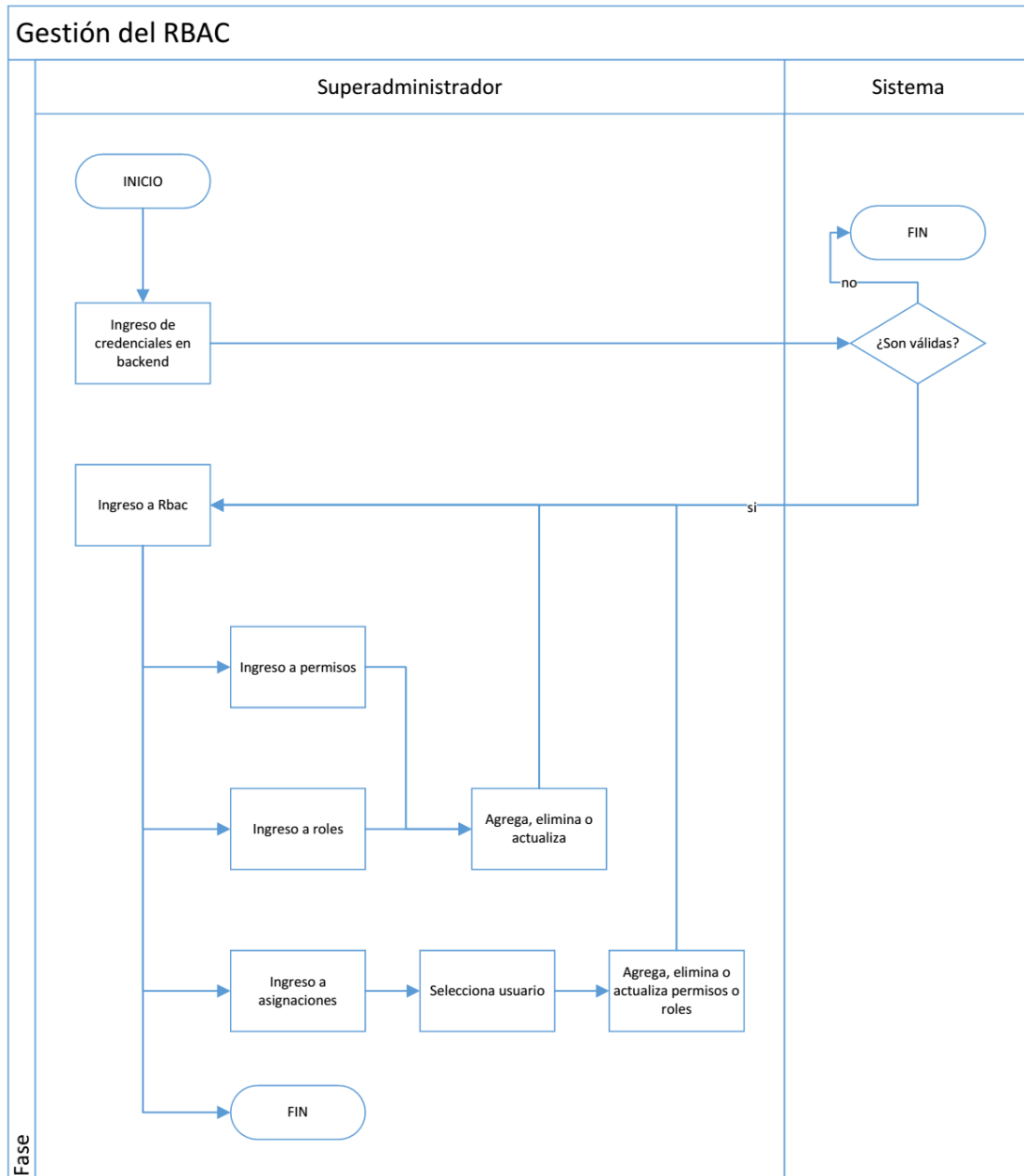


Figura 64: Diagrama de proceso gestión de los permisos para roles

Fuente: Propio

4.2.2.4. DIAGRAMA DE PROCESOS ASIGNACION DE CARTA SEMANAL A INFANTES

La Figura 65 muestra los pasos que debe realizar el usuario con rol Superadministrador para gestionar los roles.

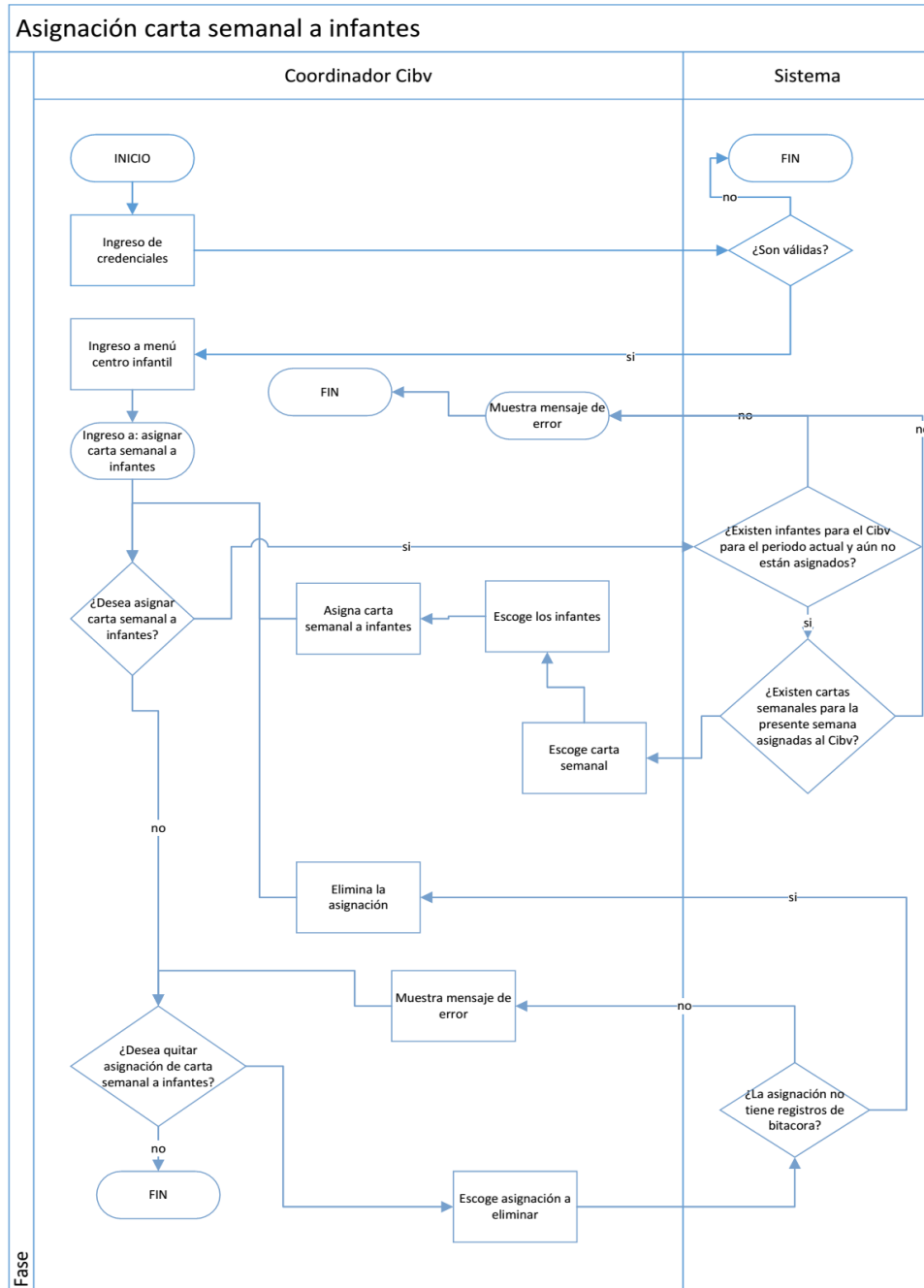


Figura 65: Diagrama de procesos para la asignación de carta semanal a infantes

Fuente: Propio

4.2.2.5. DIAGRAMA DE PROCESOS REGISTRO DE BITÁCORA

La Figura 66 muestra los pasos que debe realizar el usuario con rol Educador para gestionar la Bitácora de consumo diario infantil.

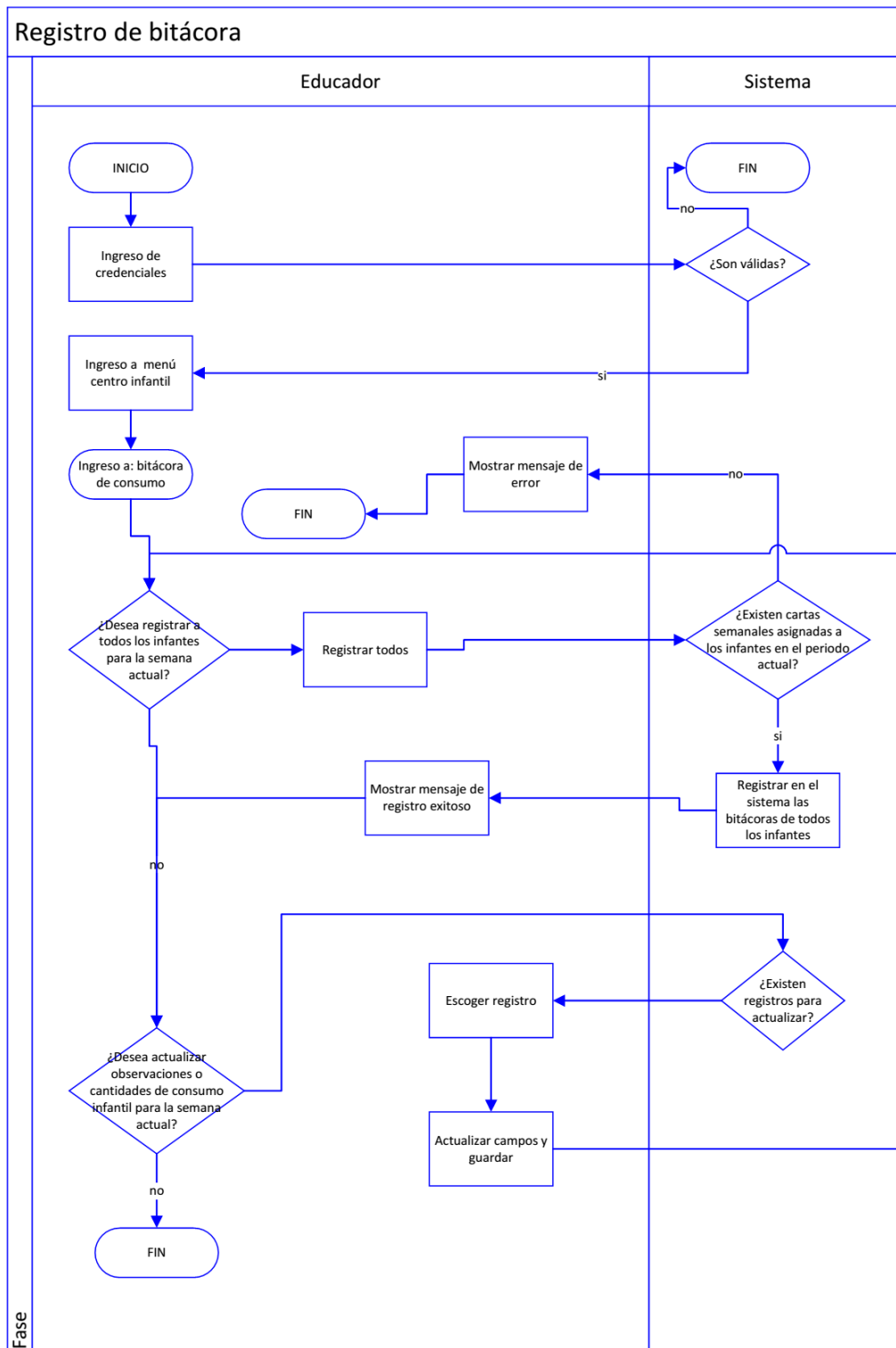


Figura 66: Diagrama de procesos registro de bitácora

Fuente:Propio

4.2.3. DIAGRAMA ENTIDAD RELACION

EL diagrama muestra la base de datos que se usó para Sinuti Fase 2.



Figura 67: Diagrama entidad relación Sinuti

Fuente: Propio

CAPÍTULO V

5. FASE DE CODIFICACIÓN

5.1.1. PLAN DE ENTREGAS

Al ordenar las historias de usuario se logró definir un plan de entregas que será cumplido en tres iteraciones una por cada módulo.

Módulos	Nro.	Historia de usuario	Fechas estimadas		Esfuerzo en desarrollo		Iteraciones			Entregas		
			Inicio	Fin	Días	Horas	1	2	3	1	2	3
Gastronomía	1	Gestion de platillos (Módulo Gastronomía)	23/11/2015	24/11/2015	2	16	x			x		
	2	Nuevo sistema de categorización para las porciones infantiles (Módulo Gastronomía)	25/11/2015	18/12/2015	18	144	x			x		
RBAC	1	Planificación Módulo RBAC	21/12/2015	25/01/2016	22	176		x			x	
Bitácora	1	Planificación módulo bitácora	26/01/2016	28/01/2016	2	16			x			x
	2	Procedimiento para porcentaje de adecuación (Módulo Bitacora)	01/02/2016	25/02/2016	19	152			x			x

Figura 68: Plan de entregas Sinuti fase 2

Fuente: Propio

5.2. FASE DE PRUEBAS

El equipo de desarrollo realiza las pruebas para el testeado de la aplicación anticipando los posibles errores del software, son definidos y escritos por el cliente en las historias de usuario. El objetivo es demostrar al cliente que el requerimiento implementado realmente funciona como el cliente lo desea (Shore & Warden, 2007).

5.2.1. PRUEBAS GESTION MÓDULO GASTRONOMÍA

La Tabla 15 muestra el caso de prueba de crear una nueva receta para el usuario con rol gastrónomo.

Tabla 15:

Caso de prueba crear nuevo receta

Prueba de aceptación			
Caso de prueba	Gestión receta	Rol de usuario	Gastrónomo
Nro. de caso de prueba	1	Nro. De historia de usuario	1
Descripción	Crear nuevo receta		

Continua 

Condiciones de ejecución	<ul style="list-style-type: none"> • El usuario deberá tener permiso para realizar esta acción • Deben de existir ingredientes o alimentos en el sistema con su respectivo análisis bromatológico • Debe de existir tipos de receta en el sistema 	
Datos de entrada	<ul style="list-style-type: none"> • Tipo receta: dato que describe el tipo de receta a preparar • Nombre: indica el nombre del receta o receta descripción: indica la descripción de la receta • Ingrediente: alimento que será usado como ingrediente de la receta • Cantidad: indica la cantidad en gramos de dicho ingrediente 	
Pasos de ejecución	<ol style="list-style-type: none"> 1. Seleccionar tipo de receta 2. Ingresar nombre y descripción de la receta 3. Seleccionar ingrediente y cantidad 4. Guardar 	
Resultado esperado	Todos los datos a ingresar son validados	SI
	Se crea la receta	SI
Evaluación	Satisfactorio	

Fuente: Propio

La Tabla 16 muestra el caso de prueba actualizar receta para el usuario con rol gastrónomo.

Tabla 16:

Caso de prueba actualizar receta

Prueba de aceptación			
Caso de prueba	Gestión receta	Rol de usuario	Gastrónomo
Nro. de caso de prueba	2	Nro. De historia de usuario	1
Descripción	Actualizar receta		
Condiciones de ejecución	<ul style="list-style-type: none"> • El usuario deberá tener permiso para realizar esta acción • Deben de existir ingredientes o alimentos en el sistema con su respectivo análisis bromatológico • Debe de existir tipos de receta en el sistema • Debe de existir el receta registrado en el sistema • No tiene que estar asignado a un menú 		
Datos de entrada	<ul style="list-style-type: none"> • Tipo receta: dato que describe el tipo de receta a preparar • Nombre: indica el nombre del receta o receta • Descripción: indica la descripción de la receta • Ingrediente: alimento que será usado como ingrediente de la receta • Cantidad: indica la cantidad en gramos de dicho ingrediente 		
Pasos de ejecución	<ol style="list-style-type: none"> 1. Seleccionar tipo de receta 2. Ingresar nombre y descripción de la receta 3. Seleccionar ingrediente y cantidad 4. Actualizar 		

Continúa



Resultado esperado	Todos los datos a actualizar son validados	SI
	Valida que ese receta este aun sin usar en los menús	SI
	Se actualiza la receta	SI
Evaluación	Satisfactorio	

Fuente: Propio

La Tabla 17 muestra el caso de prueba de eliminar receta para el usuario con rol gastrónomo.

Tabla 17:

Caso de prueba eliminar receta

Prueba de aceptación			
Caso de prueba	Gestión receta	Rol de usuario	Gastrónomo
Nro. de caso de prueba	3	Nro. De historia de usuario	1
Descripción	Eliminar receta		
Condiciones de ejecución	<ul style="list-style-type: none"> El usuario deberá tener permiso para realizar esta acción no tiene que estar asignado a un menú 		
Datos de entrada	Ninguno		
Pasos de ejecución	<ol style="list-style-type: none"> Seleccionar receta Eliminar 		
Resultado esperado	Seleccionar receta		SI
	Eliminar receta		Si
	Validación de registro en uso		SI
Evaluación	Satisfactorio		

Fuente: Propio

La Tabla 18 muestra el caso de prueba de ingresar un tipo receta para el usuario con rol gastrónomo.

Tabla 18:

Caso de prueba ingreso de tipo de receta

Prueba de aceptación			
Caso de prueba	Gestión tipo receta	Rol de usuario	Gastrónomo
Nro. de caso de prueba	4	Nro. De historia de usuario	2
Descripción	Ingreso de tipo de receta		
Condiciones de ejecución	<ul style="list-style-type: none"> Debe de existir grupos de recetas en el sistema Tener el permiso para realizar esta acción 		
Datos de entrada	<ul style="list-style-type: none"> Grupo de receta: indica el grupo de receta Tipo de receta: indica el tipo de receta a crear Nombre: indica el nombre de receta Descripción: indica la descripción de receta Cantidad: indica cantidad máxima de en gramos para el receta 		

Continúa



Pasos de ejecución	<ol style="list-style-type: none"> 1. Seleccionar grupo de receta 2. Seleccionar el equivalente al tipo de receta a crear 3. Ingresar nombre de receta 4. Ingresar descripción de receta 5. Ingresar cantidad máxima de en gramos para el receta 	
Resultado esperado	Todos los datos a ingresar son validados	SI
	Se crea el registro	Si
Evaluación	Satisfactorio	

Fuente: Propio

La Tabla 19 muestra el caso de prueba de actualizar un tipo receta para el usuario con rol gastrónomo.

Tabla 19:

Caso de prueba actualizar tipo receta

Prueba de aceptación			
Caso de prueba	Gestión tipo receta	Rol de usuario	Gastrónomo
Nro. de caso de prueba	5	Nro. De historia de usuario	2
Descripción	Actualización de tipo de receta		
Condiciones de ejecución	<ul style="list-style-type: none"> • Debe de existir grupos de recetas en el sistema • tener el permiso para realizar esta acción • Que ese tipo de preparación no esté en uso en una receta 		
Datos de entrada	<ul style="list-style-type: none"> • Grupo de receta: indica el grupo de receta • Tipo de receta: indica el tipo de receta a crear • Nombre: indica el nombre de receta • Descripción: indica la descripción de receta • Cantidad: indica cantidad máxima de en gramos para el receta 		
Pasos de ejecución	<ol style="list-style-type: none"> 1. Seleccionar tipo de receta a actualizar 2. Seleccionar grupo de receta 3. Seleccionar el equivalente al tipo de receta a crear 4. Ingresar nombre de receta 5. Ingresar descripción de receta 6. Ingresar cantidad máxima en gramos para el receta 		
Resultado esperado	Seleccionar receta	SI	
	Actualización	Si	
	Validación de registro en uso	SI	
Evaluación	Satisfactorio		

Fuente: Propio

La Tabla 20 muestra el caso de prueba de eliminar un tipo receta para el usuario con rol gastrónomo.

Tabla 20:
Caso de prueba eliminar tipo receta

Prueba de aceptación			
Caso de prueba	Gestión tipo receta	Rol de usuario	Gastrónomo
Nro. de caso de prueba	6	Nro. De historia de usuario	2
Descripción	Eliminar tipo de receta		
Condiciones de ejecución	<ul style="list-style-type: none"> Tener el permiso para realizar esta acción Exista el tipo de receta a eliminar Que no se elimine el tipo de receta que está en uso en los recetas 		
Datos de entrada	Ninguno		
Pasos de ejecución	<ol style="list-style-type: none"> Seleccionar tipo de receta Eliminar 		
Resultado esperado	Seleccionar receta	SI	
	Validación de eliminación	Si	
	Eliminación	SI	
Evaluación	Satisfactorio		

Fuente: Propio

5.2.2.PRUEBAS GESTIÓN MÓDULO RBAC

La Tabla 21 muestra el caso de prueba crear un nuevo permiso y actualizarlo para el usuario con rol Superadministrador.

Tabla 21:
Caso de prueba crear nuevo permiso y actualizarlo

Prueba de aceptación			
Caso de prueba	Gestión Rbac	Rol de usuario	Superadministrador
Nro. de caso de prueba	6	Nro. De historia de usuario	3
Descripción	Crear nuevo permiso y actualizarlo		
Condiciones de ejecución	<ul style="list-style-type: none"> El usuario deberá tener permiso para realizar esta acción 		
Datos de entrada	<ul style="list-style-type: none"> Nombre: el nombre del permiso Descripción: la descripción del permiso Nombre de regla: Indica cual regla va a ser usada para ese permiso Datos: datos extra para el funcionamiento de la regla 		

Continúa 

Pasos de ejecución	<ol style="list-style-type: none"> 1. Ingresar nombre 2. Ingresar descripción 3. Ingresar regla 4. Ingresar datos 5. Guardar o actualizar 	
Resultado esperado	Todos los datos a ingresar son validados	SI
	Se crea el permiso	SI
	Se actualiza el permiso	SI
Evaluación	Satisfactorio	

Fuente: Propio

La Tabla 22 muestra el caso de prueba eliminar permiso para el usuario con rol Superadministrador.

Tabla 22:
Caso de prueba eliminar permiso


Prueba de aceptación			
Caso de prueba	Gestión Rbac	Rol de usuario	Superadministrador
Nro. de caso de prueba	7	Nro. De historia de usuario	3
Descripción	Eliminar permiso		
Condiciones de ejecución	<ul style="list-style-type: none"> • El usuario deberá tener permiso para realizar esta acción • Debe de existir una regla para ser eliminada 		
Datos de entrada	Ninguno		
Pasos de ejecución	<ol style="list-style-type: none"> 1. Seleccionar permiso 2. Eliminar 		
Resultado esperado	Se elimina el permiso	SI	
Evaluación	Satisfactorio		

Fuente: Propio

La Tabla 23 muestra el caso de prueba crear un nuevo rol y actualizarlo para el usuario con rol Superadministrador.

Tabla 23:
Caso de prueba crear nuevo rol y actualizarlo

Prueba de aceptación			
Caso de prueba	Gestión Rbac	Rol de usuario	Superadministrador
Nro. de caso de prueba	8	Nro. De historia de usuario	3
Descripción	Crear nuevo rol y actualizarlo		
Condiciones de ejecución	<ul style="list-style-type: none"> • El usuario deberá tener permiso para realizar esta acción 		
Datos de entrada	<ul style="list-style-type: none"> • Nombre: el nombre del rol • Descripción: la descripción del rol • Nombre de regla: Indica cual regla va a ser usada para ese rol • Datos: datos extra para el funcionamiento de la regla 		

Continúa 

Pasos de ejecución	<ol style="list-style-type: none"> 1. Ingresar nombre 2. Ingresar descripción 3. Ingresar regla 4. Ingresar datos 5. Guardar o actualizar 	
Resultado esperado	Todos los datos a ingresar son validados	SI
	Se crea el rol	SI
	Se actualiza el rol	SI
Evaluación	Satisfactorio	

Fuente: Propio

La Tabla 24 muestra el caso de prueba eliminar rol para el usuario con rol Superadministrador.

Tabla 24:

Caso de prueba eliminar permiso

Prueba de aceptación			
Caso de prueba	Gestión Rbac	Rol de usuario	Superadministrador
Nro. de caso de prueba	9	Nro. De historia de usuario	3
Descripción	Eliminar rol		
Condiciones de ejecución	<ul style="list-style-type: none"> • El usuario deberá tener permiso para realizar esta acción • Debe de existir un rol para ser eliminado 		
Datos de entrada	Ninguno		
Pasos de ejecución	<ol style="list-style-type: none"> 1. Seleccionar rol 2. Eliminar 		
Resultado esperado	Se elimina el permiso		SI
Evaluación	Satisfactorio		


Fuente: Propio

5.2.3. PRUEBAS GESTION MÓDULO BITÁCORA

La Tabla 25 muestra el caso de asignar carta semanal a infantes para el usuario con rol Coordinador.

Tabla 25:

Caso de prueba asignar carta semanal a infantes

Prueba de aceptación			
Caso de prueba	Gestión asignación bitácora	Rol de usuario	Coordinador
Nro. de caso de prueba	11	Nro. De historia de usuario	4
Descripción	Asignar carta semanal a i		
	Continua		

Condiciones de ejecución	<ul style="list-style-type: none"> • El usuario deberá tener permiso para realizar esta acción • Deberá de tener infantes registrados al cibv al cual el usuario de rol coordinador pertenezca en el periodo actual • Deberá de tener registrada por lo menos una carta semanal para ese cibv 	
Datos de entrada	<ul style="list-style-type: none"> • Lista Infantes: nombres de los infantes • Carta semanal: carta semanal disponible para consumo • Observaciones: Observaciones de la asignación 	
Pasos de ejecución	<ol style="list-style-type: none"> 1. Seleccionar infantes 2. Seleccionar carta semanal 3. Ingresar observaciones de la asignación 4. Guardar 	
Resultado esperado	Todos los datos a ingresar son validados	
	Se valida que el límite de comensales de la carta semanal no se exceda	
	No repetir infantes en la asignación	
	Se guarda	
Evaluación		

Fuente: Propio

CAPÍTULO VI

6. ANÁLISIS DE IMPACTOS

6.1. ANÁLISIS COMPARATIVO DEL FRAMEWORK YII 2 VS YII 1.1.16

Los procesos de análisis de datos ayudan a identificar los datos útiles con el propósito de sacar conclusiones que permiten tomar mejores decisiones. Dentro de estos procesos se encuentra el análisis confirmatorio de datos que a través de indicadores confirman o no una hipótesis (Olivares, 2016).

El presente estudio midió la eficiencia del ORM y velocidad de procesamiento de las peticiones con herramientas de benchmarking de código abierto para frameworks PHP. Estas se encuentran en repositorio GitHub.

La primera denominada PHP framework Benchmark es un proyecto que intenta medir la sobrecarga mínima de los frameworks PHP sin que se usen plantillas Html, manipulación de base de datos o información de depuración. El proyecto inicialmente cuenta con 16 distintos frameworks entre los cuales se encuentra Yii 2 y a este se le agrego Yii 1.1.6 (kenjis, PHP Framework Benchmark, 2016).

La segunda (PHP ORM Benchmark) herramienta consiste en generar un script y evaluar la eficiencia al obtener datos con el ORM de cada versión de Yii (kenjis, PHP ORM Benchmark, 2016).

El anexo D provee de información sobre las herramientas y configuraciones del mismo.

La Figura 69 muestra los resultados obtenidos del benchmarking evaluando la eficiencia de procesamiento de las peticiones.

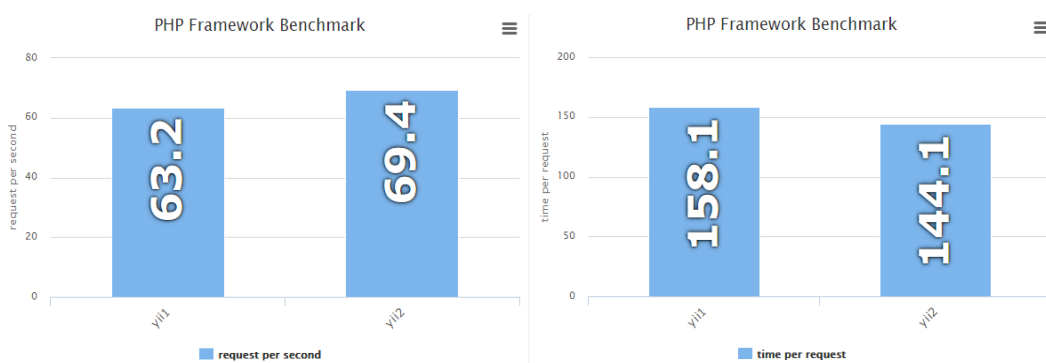


Figura 69: Resultados de eficiencia en procesamiento de peticiones

Fuente: Propio

La Figura 70 muestra los resultados obtenidos del benchmarking evaluando la eficiencia del ORM.

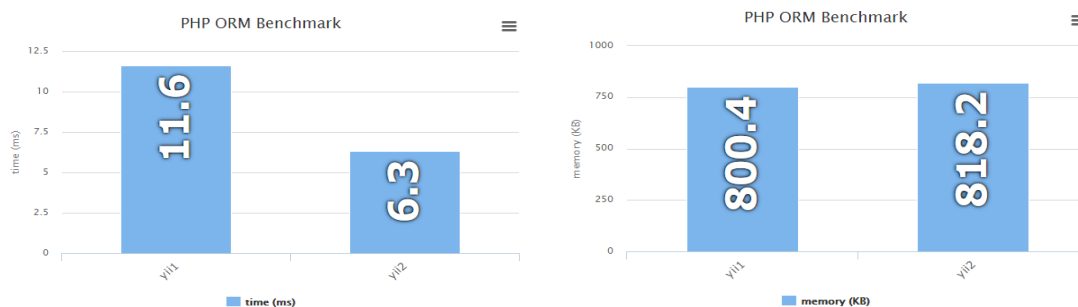


Figura 70: Resultado de la eficiencia del ORM

Fuente: (kenjis, PHP ORM Benchmark, 2016)

La Tabla 26 indica que la versión 2 de Yii posee mejores características en los aspectos de procesamiento de las peticiones, la versión 1 consume menos memoria al ejecutar una consulta usando el ORM pero requiere de más tiempo de procesamiento de la misma que en la versión 2.

Tabla 26:
Resultados benchmarking frameworks PHP

CARACTERÍSTICAS	YII 1.1.6	YII 2.0.6
Cantidad promedio de peticiones por segundo	63.2	69.4
Tiempo promedio que se tomó en ejecutar cada request	158.1	144.1
Tiempo (ms) promedio en ejecutar la consulta	11.6	6.3
Cantidad promedio de consumo de la memoria de la consulta	800.4	818.2

Fuente: Propio

6.2. ANÁLISIS DE IMPACTOS EN EL SERVICIO DE LOS CIBV DEL SISTEMA NUTRIMENTAL INFANTIL FASE DOS

Los impactos son los aspectos positivos o negativos que la ejecución del proyecto provocó en un ámbito determinado. El desarrollo de la Fase 2 de Sinuti generó los siguientes impactos:

Aspecto tecnológico

El aplicativo mejoró el sistema Sinuti con respecto al seguimiento nutricional y alimenticio de los infantes, esto permite que se tengan indicadores precisos e individuales sobre el crecimiento de los niños. Además, automatizó los procedimientos de: preparación, búsqueda

y modificación de recetas que inciden directamente sobre los responsables del área de preparación de alimentos.

El sistema permitió disminuir el tiempo de espera para obtener el análisis bromatológico de las recetas ya que el sistema cuenta con un catálogo de alimentos con sus respectivos valores nutricionales y automatizó las operaciones matemáticas para la cantidad que el gastrónomo requiera.

El sistema optimizó el tiempo gracias a los registros de bitácora de consumo que son generados con este, permitiendo al nutricionista tener indicadores de alimentación precisa sobre lo consumido por el infante. También mejoró los indicadores sobre el porcentaje de consumo ideal y que tipo de nutrientes debe consumir un infante en específico para mantenerse saludable.

Aspecto económico

El sistema ayudó a precisar el procedimiento matemático para conocer con exactitud la cantidad de cada valor nutricional de una receta. Por lo cual se puede hacer simil al análisis bromatológico de un laboratorio.

La consulta y el tiempo necesarios para disponer de los servicios de un nutricionista infantil llegan a costar alrededor de \$ 100, debido a los procedimientos manuales que requiere hacer relacionando la alimentación con la edad del niño. El sistema realizó estas tareas con procedimientos matemáticos, en menor tiempo y muestra los resultados de manera comprensible al usuario.

Aspecto ambiental

El proceso para la generación, guardado y búsqueda de recetas, se lo realizaba manualmente, además la tarea de conocer el porcentaje de adecuación alimenticia para el niño a través de la planificación semanal de alimentación con la ficha del niño llevaba mucho tiempo, por lo que la implementación del sistema permitió reducir el uso de papel y disminuir los tiempos de respuesta.

CONCLUSIONES

- Al implementar el Control de Acceso Baso en Roles (Rbac) a Sinuti, se agregó las funcionalidades de gestión de usuarios, roles y permisos a todo el sistema. Esto permite la mejora en la escalabilidad en la administración de usuarios y evita que los usuarios realicen tareas no asignadas.
- El automatizar los procesos con los cuales se crea una receta, permitió conocer la cantidad necesaria de alimentos que estas deben contener para que no existan excedentes en la alimentación de los infantes. También permitió generar un recetario digital el cual reemplazo al recetario en papel que anteriormente se manejaba.
- La bitácora de consumo infantil ayudó al nutricionista a conocer cuáles son los requerimientos de alimentación infantiles por cada individuo, lo cual ayuda a realizar un diagnóstico más completo y exacto.
- En cuanto al framework, se concluye que Yii 2 es más eficiente en cuanto a procesamiento de peticiones y manejo de datos que Yii 1. Además, este es uno de los frameworks más completos dentro del área del desarrollo para el lenguaje PHP.
- Realizar presentaciones y tests en la fase de desarrollo y producción del software con los interesados, permitió que el proyecto pueda ser corregido a tiempo a partir de: las observaciones de los clientes y problemas de configuración del framework con el sistema de alojamiento, ayudando a la finalización con éxito del proyecto.

RECOMENDACIONES

- Es recomendable usar sistemas para el control de versiones que permiten supervisar y mantener un historial de cada etapa del desarrollo del software. El control de versiones permite que el equipo de desarrollo disponga de manera inmediata de los cambios realizados en el código por sus compañeros.
- Es recomendable usar un host virtual para simular un alojamiento en un servidor dedicado y que las configuraciones que se realizan sobre la mayoría de archivos para la redirección deberán de ser probadas en producción sobre el espacio de alojamiento que el cliente disponga.
- En la fase de desarrollo, es recomendable que se desarrolle el entorno de API RESTful como una aplicación separada, diferente del Frontend y Backend, con el fin de facilitar el mantenimiento del código. Posteriormente se la deberá de agregar al proyecto que se publicará.
- Se recomienda usar los métodos del creador de queries Command Query debido a que Yii no dispone de funciones en específico que ayuden a realizar consultas SQL avanzadas.

BIBLIOGRAFÍA

- Acens. (28 de 01 de 2016). Obtenido de Framework para el desarrollo de aplicaciones ágiles: <http://www.acens.com/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf>
- ALEGSA. (06 de 02 de 2016). *Definición de widget (gui)*. Obtenido de <http://www.alegsa.com.ar/Dic/widget.php>
- Aravind Shenoy, U. S. (2014). *Learning Bootstrap*. Birmingham: Packt Publishing.
- Brandt, C., Klimov, P., Domba Cerin, M., Xue, Q., Makarov, A., & Naumenko, D. (02 de 05 de 2016). *Yii 2 Advanced Project Template*. Obtenido de <https://github.com/yiisoft/yii2-app-advanced/tree/master/docs/guide>
- Caro, P. S. (09 de 02 de 2016). *Unified Modeling Language (Tutorial)*. Obtenido de <http://users.dcc.uchile.cl/~psalinas/uml/>
- David. (30 de 03 de 2016). *¿ Hoy en día es rentable vivir del desarrollo de páginas web ?* Obtenido de <http://www.forosdelweb.com/f91/hoy-dia-rentable-vivir-del-desarrollo-paginas-web-1031626/>
- Dirección Nacional de Servicios Centros Infantiles del Buen Vivir. (2013). *Guía Teórica Metodológica. Guía Teórica - Metodológica CIBV*. Quito.
- Fulcher, S. (21 de 05 de 2012). *The 3-Layer Architecture vs. the CMS*. Obtenido de <https://www.redweb.com/agency/blog/2012/june/3-layer-architecture-vs-cms>
- Foundation, A. S. (16 de 05 de 2016). *Apache HTTP Server Tutorial: .htaccess files*. Obtenido de <https://httpd.apache.org/docs/current/howto/htaccess.html>
- GAD-I. (2015). *Convenio de cooperación económica entre el MIES y GAD-Ibarra para la implementación de servicios de desarrollo infantil*. Ibarra.
- Irawan, B. (18 de 07 de 2014). *Setup RESTful API in Yii2*. Obtenido de <http://budiirawan.com/setup-restful-api-yii2/>
- kenjis. (25 de 03 de 2016). *PHP Framework Benchmark*. Obtenido de <https://github.com/kenjis/php-framework-benchmark/tree/optimize>
- kenjis. (19 de 02 de 2016). *PHP ORM Benchmark*. Obtenido de <https://github.com/kenjis/php-orm-benchmark>
- Klimov, P. (15 de 02 de 2016). *HTTP Client Extension for Yii 2*. Obtenido de <https://github.com/yiisoft/yii2-httpclient>
- LLC, Y. S. (25 de 03 de 2016). *Acerca de Yii*. Obtenido de <http://www.yiiframework.com/about/>
- MIES. (2014). *Norma Técnica de Desarrollo Infantil Integral*. Quito: (D. d. MIES, Ed.).
- Ministerio Coordinador de Desarrollo Social. (12 de Noviembre de 2015). *Programa Acción Nutrición*. Obtenido de <http://www.desarrollosocial.gob.ec/programa-accion-nutricion/>
- Munir, M. D. (26 de 04 de 2016). *RBAC Manager for Yii 2*. Obtenido de <https://github.com/mdmsoft/yii2-admin>
- Olivares, J. (18 de 02 de 2016). *uaem redalyc*. Obtenido de <http://www.redalyc.org/articulo.oa?id=12211106>
- Oracle. (04 de 02 de 2016). *Guía de administración del sistema: servicios de seguridad*. Obtenido de https://docs.oracle.com/cd/E24842_01/html/E23286/rbac-1.html
- PÉREZ, A. (18 de 02 de 2016). *Configuración MySQL Master Slave*. Obtenido de <http://helloit.es/2013/02/configuracion-mysql-master-slave/>
- PHP-Group. (02 de 18 de 2016). *PHP Manual*. Obtenido de <http://ir1.php.net/manual/en/>
- Pitt, C. (2012). *Pro PHP MVC*. Apress.

- Podlinov, R. (18 de 12 de 2014). *What the idea behind environment folders in Yii2 and how to use it?* Obtenido de <http://stackoverflow.com/questions/27542537/what-the-idea-behind-environment-folders-in-yii2-and-how-to-use-it#9490>
- Popel, D. (2007). *Learning PHP Data Objects*. Birmingham: Packt Publishing Ltd.
- Ramírez, F. (2012). *Aprénda practicando ASP.NET usando visual studio 2012*. México: Alfaomega Grupo Editor.
- Rondón Castaño, J. F. (30 de 03 de 2016). *Importancia del desarrollo de software*. Obtenido de http://www.academia.edu/8842504/IMPORTANCIA_DEL_DESARROLLO_DE_SOFTWARE
- Sanchez, X. (30 de 03 de 2016). *¿DEBERÍAS USAR UN FRAMEWORK PARA TU PROYECTO WEB?* Obtenido de <http://www.emprenderalia.com/deberias-usar-un-framework-para-tu-proyecto-web/>
- Shore, J., & Warden, S. (2007). *The art of agile development*. Sebastopol: Mary O'Brien.
- Somoza, J. M. (16 de 02 de 2016). *Ventajas y desventajas de los CMS (I) - Joomla*. Obtenido de <http://www.pymenetworks.es/blog/ventajas-y-desventajas-de-los-cms-i-joomla>
- Swedberg, J. C. (2013). *Learning Jquery - Fourth Edition*. Pack Pub .
- Tangent-LLC. (07 de 02 de 2016). *Metodología XP*. Obtenido de <https://procesosdesoftware.wikispaces.com/METODOLOGIA+XP>
- Twin-Coders. (18 de 02 de 2016). *Viajando en el tiempo con pushState()*. Obtenido de <http://twincoders.com/blog/viajando-en-el-tiempo-con-pushstate/>
- Vahrmeijer, B. (20 de Agosto de 2014). *Definition of Switching point - Frontend vs Backend Issue #4771 · yiisoft/yii2 · GitHub*. Obtenido de <https://github.com/yiisoft/yii2/issues/4771>
- Vidal, G. (09 de 02 de 2016). *Ingeniería de requerimientos*. Obtenido de http://users.dsic.upv.es/~jsilva/fin/idr/IDR_practica1.pdf
- Wanstrath, C. (14 de 01 de 2016). *Pjax = pushState + ajax*. Obtenido de <https://github.com/defunkt/jquery-pjax/blob/master/README.md>
- Xue, Q., Makarov, A., Brandt, C., Klimov, P., & Y. c. (2014). *The Definitive Guide to Yii 2.0*. Birmingham: Yii Software LLC.
- Yii Software LLC. (14 de Noviembre de 2015). *Yii PHP Framework Version 2*. Obtenido de <https://github.com/yiisoft/yii2>
- Yii Supporters. (12 de Noviembre de 2015). *Yii Framework 2.0 API Documentation*. Obtenido de <http://www.yiiframework.com/doc-2.0/>
- Yii-Suporters. (06 de 02 de 2016). *La guía definitiva para Yii*. Obtenido de <http://www.yiiframework.com/doc/guide/1.1/es/>

ANEXOS

ANEXOS

Los anexos descritos los puede encontrar en la carpeta Anexos que se encuentra dentro del CD del presente proyecto

ANEXO A: INSTALACIÓN DE UN THEME PARA FRONTEND (EN CD)

ANEXO B: USO DE OBJETOS DE PARA MANEJO DEL MODELO EN YII 2 (EN CD)

ANEXO C: ESPECIFICACION DE REQUERIMIENTOS

ANEXO D: PRUEBAS DE ACEPTACIÓN

ANEXO E: BENCHMARKING PHP FRAMEWORKS (EN CD)

ANEXO F: HISTORIAS DE USUARIO

ANEXO G: MANUAL TÉCNICO (EN CD)

ANEXO H: MANUALES DE USUARIO (EN CD)

GLOSARIO

¹ Css3: Lenguaje de hoja de estilo que describe la presentación de un HTML (o XML). Fuente: <http://www.w3schools.com/css/>.

² HTML5: Es un lenguaje de marcado para la descripción de documentos web (páginas web). Fuente: <http://www.w3schools.com/html/>.

³ PHP: Lenguaje de script de servidor, y una herramienta de gran alcance para hacer páginas web dinámicas e interactivas (Popel, 2007).

⁴ Composer: Herramienta para la gestión de dependencias para PHP. Fuente: <https://getcomposer.org/>.

⁵ RESTful API: Provee de funcionalidades de creación, actualización, lectura y eliminado de registros a través de web services tipo REST <http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>.

⁶ NoSQL: Son un tipo de base de datos que se caracterizan por su alta velocidad en las lecturas de datos y su poca o escasa confiabilidad en la consistencia de los mismos. Fuente: <https://es.wikipedia.org/wiki/NoSQL>.

⁷ CMS: Sistema de gestión de contenidos es un programa que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web con los roles necesarios para su correcta administración. Fuente: https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos.

⁸ e-commerce: Consiste en la compra y venta de productos y servicios a través de medios electrónicos. Fuente: https://es.wikipedia.org/wiki/Comercio_electr%C3%B3nico.

⁹ CRM: Customer Relationship Management es una aplicación que permite la gestión comercial, de marketing y atención al cliente. Fuente: https://es.wikipedia.org/wiki/Customer_relationship_management.

¹⁰ ERP: Enterprise Resource Planning es una aplicación que se encuentra orientado hacia áreas operativas de la empresa, tales como Finanzas, Recursos Humanos, Producción. Fuente: <https://es.wikipedia.org/wiki/ERP>.

¹¹ RBAC: El control de acceso basado en roles (RBAC) es una función de seguridad para controlar el acceso de usuarios a tareas que normalmente están restringidas al superusuario. Mediante la aplicación de atributos de seguridad a procesos y usuarios. Fuente: http://docs.oracle.com/cd/E24842_01/html/E23286/rbac-1.html.

¹² JQuery: Es una librería de Javascript que potencia la manipulación de HTML a través de sencillas funciones. Fuente: <https://jquery.com/>.

¹³ WAMP: Es un entorno de desarrollo configurado en base a Windows, Apache, MySQL y PHP con una arquitectura cliente servidor. Fuente: <http://www.wampserver.com/en/>.

¹⁴ Centos: Es una distribución Linux de un sistema operativo para servidores. Fuente: <https://www.centos.org>.

¹⁵ LAMP: Es un entorno de desarrollo configurado en base a Linux, Apache MySQL y PHP. Fuente: <https://es.wikipedia.org/wiki/LAMP>.

¹⁶ Cibv: Centro(s) infantil(es) del buen vivir. Fuente: (Dirección Nacional de Servicios Centros Infantiles del Buen Vivir, 2013).

¹⁷ Google: Compañía especializada en productos y servicios relacionados con Internet. Fuente: <https://es.wikipedia.org/wiki/Google>.

¹⁸ Start-Up: Es un término utilizado actualmente en el mundo empresarial que traduce arrancar o emprender. Fuente: https://es.wikipedia.org/wiki/Compa%C3%B1a_startup.

¹⁹ Mies: Ministerio de inclusión económica y social, el cual procura generar oportunidades de trabajo y de buen vivir aplicando las políticas del estado ecuatoriano. Fuente: <http://www.inclusion.gob.ec/nuevo-mies/>.

²⁰ Ieps: Instituto de economía popular y solidaria, esta busca la inclusión de todos los ciudadanos en ámbitos financieros, sociales, culturales y políticos. Fuente: <http://www.economiasolidaria.gob.ec/la-institucion/>.

²¹ Magap: Ministerio de Agricultura, Ganadería, Acuicultura y Pesca, está encargado de coordinar, diseñar y evaluar las políticas del sector agropecuario con el resto de sectores económicos y sociales. Fuente: http://ecuadorecologico.com/directorio_empresas_ambientales_ecuador/ministerio-de-agricultura-ganaderia-acuicultura-y-pesca/.

²² XP: Programación extrema es una metodología ágil de desarrollo de software exige el testeo continuo, refactorización continua y siempre mantener una relación con el cliente. Fuente: (Tangient-LLC, 2016).

²³ Scrum: Es un proceso de desarrollo iterativo de software, este no exige de una planificación inicial de desarrollo o entregas y exige de auto organización del equipo. Fuente: <https://proyectosagiles.org/que-es-scrum/>.

²⁴ MVC: Modelo Vista Controlador, es un patrón de arquitectura que separa datos, lógica de negocio e interfaz del usuario. Fuente: (Yii Suporters, 2015).

⁴⁷ CRUD: Acrónimo en inglés de: crear, leer, actualizar y eliminar usado para referirse a las funciones básicas de la base de datos. Fuente: (Pitt, 2012).

⁴⁸ Ajax: acrónimo de Asynchronous JavaScript And XML, permite crear aplicaciones dinámicas e interactivas sin recargar las páginas de los navegadores web ya que se comunican en forma asíncrona con el servidor. Fuente: <https://es.wikipedia.org/wiki/AJAX>.

⁴⁹ ACF : Filtros de control de acceso, estos proveen de seguridad básica para la aplicación, en Yii 2 tiene por permisos de acceso para usuario autenticados e invitados. Fuente: (Xue, Makarov, Brandt, Klimov, & Yii, 2014).

⁵⁰ PushState: Método de HTML5 para añadir entradas al historial, permite que se añadan contenidos dinámicos a las páginas web y que estos tengan su propio historial para ser accedidos desde la url del navegador. Fuente: <http://blog.alebanzas.com.ar/2011/11/html5-pushstate-no-mas-location-hash/>.

⁵¹ IEEE: Corresponde a las siglas en inglés de "Instituto de Ingenieros Eléctricos y Electrónicos", es una asociación técnico-profesional de carácter mundial dedicada a la estandarización de las tecnologías. Fuente: <https://www.ieee.org/index.html>.

⁵² Stakeholders: Término en inglés que hace referencia a las personas, organizaciones y empresas que se vinculan en el desarrollo del software. Fuente: (Shore & Warden, 2007).

⁵³ Git: Es un software de control de versiones que permite la gestión de los diversos cambios sobre el código. Fuente: <https://git-scm.com/>.

⁵⁴ BitBucket: Servicio de alojamiento en la web para los proyectos que usan control de versiones como Git. Fuente: <https://bitbucket.org/>.

⁵⁵ Endpoint: Provee de información necesarias para acceder y entablar comunicación con servicios web. Fuente: <http://di002.edv.uniovi.es/~falvarez/WSDL.pdf>.

⁵⁶ UML: (Lenguaje de Modelado Unificado) es un estándar internacional para crear esquemas, diagramas y documentación relativa al desarrollo de software. Fuente: (Caro, 2016).