



DISEÑO DETALLADO DEL SOFTWARE

PdswProject

Autores:

Jhordy Esteban Salinas Santiago

Carlos Andrés Sánchez Venegas

Leonardo Andrés Herrera

Co-autor/Asesor: Profesor Héctor Cadavid R.

PREFACIO

Este documento describe el diseño detallado de software del módulo PdsWProject, cuyo objetivo principal es automatizar la entrega de equipos en el laboratorio de electrónica

Alcance

Este documento de diseño detallado del software del módulo es la base de la codificación de este. Describe los siguientes tópicos: Definición del Software, Diseño Detallado, Descripción del proceso, Glosario (Definiciones y Siglas). Este documento no describe riesgos relacionados con el modulo que puedan impactar el desarrollo, Arquitectura del módulo,(librerías, semáforos, mutex, colas de mensaje, variables globales, etc.), las macros privadas que son usadas internamente por el módulo, las constantes privadas que son utilizadas por el módulo (identificador, valor, descripción), los tipos de datos enumerados privados que son usados internamente, las estructuras de datos privadas que son usadas internamente, las variables privadas que son usadas internamente (identificador, valor, descripción), las rutinas privadas que serán implementadas entregando una breve descripción de su función, las rutinas públicas que serán implementadas que requieren un tratamiento especial o cuyo diseño no es obvio

HISTORIA DEL DOCUMENTO

Fecha	Versión	Comentarios	Autor
20 Mayo 2016	0.1	Versión inicial (más o menos mitad)	Equipo
21 Mayo 2016	1.0	Agrego diagrama presentación	Carlos Sánchez
22 Mayo 2016	2.0	Agrego diagrama lógica	Jhordy Salinas
23 Mayo 2016	3.0	Agrego diagrama persistencia	Leonardo Herrera
23 Mayo 2016	4.0	Agrego diagrama de caja	Leonardo Herrera
23 Mayo 2016	5.0	Agrego diagramas restantes (scrum), glosario y algunos numerales faltantes	Jhordy Salinas

TABLA DE CONTENIDOS

1	Definición del Software.....	1
1.1	Visión del producto	1
1.2	Descripción del contexto en el cual se utilizará el producto.....	1
1.3	Historias de usuario	1
1.4	Alcance	3
1.5	Supuestos.....	3
1.6	Dependencias.....	3
1.7	Metodología de Prueba	4
1.8	Estándares seguidos	5
2	Diseño Detallado	6
2.1	Arquitectura de alto nivel	6
2.2	Modelo relacional.....	7
2.3	Diseño por paquetes/capas	8
2.3.1	Capa de persistencia.....	8
2.3.2	Capa lógica	9
2.3.3	Capa de presentación	12
3	Descripción del proceso	17
3.1	Descripción del proceso de desarrollo utilizado, y de las prácticas consideradas para el mismo.....	17
3.2	<i>Backlogs</i> y <i>Burndown-charts</i> de los Sprints.....	17
3.3	Release-Burndown-chart	18
3.4	Métricas por desarrollador	20
	Glosario (Definiciones y Siglas)	22

1 Definición del Software

1.1 Visión del producto

Actualmente el laboratorio de Ingeniería Electrónica presta un importante servicio a la institución al facilitar equipos electrónicos de última tecnología -con fines académicos y de investigación aplicada- a toda la comunidad de la institución. El laboratorio cuenta con una moderna plataforma tecnológica que permite llevar tanto el control del uso adecuado de los equipos, como el aplicar más fácilmente las políticas de renovación y actualización de los mismos

Gracias a esta plataforma, el proceso de préstamo de equipos resulta sumamente simple, tanto para el Laboratorista como para los estudiantes. Haciendo uso de las codificaciones de códigos de barras disponibles en los carnets de los estudiantes y en los códigos de inventario de los equipos, la incorporación del lector de código de barras reduce el proceso de préstamo a la lectura del código del estudiante y a la posterior lectura de los códigos de los equipos que se van a prestar. De la misma manera, el proceso de devolución es sumamente simple, pues haciendo uso de estos mismos códigos el sistema rápidamente identifica a quien debe descargar el equipo cuyo código se lee al momento de la devolución.

Por otro lado, esta plataforma permite aplicar más fácilmente el reglamento del laboratorio, al ofrecer de forma inmediata y actualizada, información de qué equipos se encuentran en mora de ser regresados, o qué equipos han cumplido ya -por ejemplo- con un número de horas de uso para realizar su mantenimiento o darlo de baja.

1.2 Descripción del contexto en el cual se utilizará el producto

El producto software será usado en la gestión de los equipos que se encuentran en el laboratorio de electrónica de la Escuela Colombiana de Ingeniería “Julio Garavito”, automatizando procesos realizados a la hora de registrar modelos, equipos, herramientas, préstamos, devoluciones y diferentes consultas para el control de estos.

1.3 Historias de usuario

NOMBRE	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN
Registrar una devolución	Como Laboratorista Quiero poder registrar la devolución de uno o más equipos Para dejar el estado de cuenta del	Si el usuario va a regresar elementos codificados, debe bastar con que en el sistema se ingrese el código del equipo. Una vez hecho esto, el sistema debe mostrar: nombre del usuario que realiza la devolución, y confirmar que el equipo ha sido descargado de los préstamos de dicho usuario

	usuario en limpio	Si el usuario va a regresar elementos no codificados, debe ingresar el código del estudiante, seleccionar el tipo de equipo, e ingresar el número de ítems que se está regresando.
Registrar equipo en el inventario	Como Laboratorista Quiero poder registrar un nuevo equipo en el inventario Para dejarlo disponible para préstamo y que los estudiantes, profesores e investigadores puedan hacer uso del mismo.	Antes de registrar el equipo, el sistema debe permitir buscar si el modelo de dicho equipo ya está registrado en el sistema, y si lo está, agregarlo como un nuevo elemento del mismo.
		Si el equipo a registrarse corresponde a un modelo que aún no está registrado, se le debe solicitar al usuario el ingreso de los detalles del modelo, incluyendo: Modelo, nombre, clase, vida útil en horas, valor comercial aproximado, si está asegurado o no y su fotografía (esta última puede ser una fotografía disponible libremente en Internet)
		El sistema no debe permitir que se registre dos veces el mismo modelo, ni que se registre dos veces el mismo serial/placa asociada al mismo equipo.
Registrar un préstamo	Como Laboratorista Quiero poder registrar el préstamo de uno o varios equipos Para poder ofrecer temporalmente el equipo a un estudiante, llevando control de devolución del mismo.	El sistema debe primero solicitar la identificación del estudiante/profesor/investigador, y permitir ingresarla manualmente o mediante un lector de código de barras (con el carnet del estudiante). Para esto, el foco de la interfaz debe estar inicialmente en el campo que permite ingresar dicho código.
		Una vez se identifique el estudiante/profesor/investigador, el sistema debe permitir ingresar de la misma manera (con teclado o código de barras) los códigos de uno o más equipos. Para cada equipo se debe indicar el tipo de préstamo a realizar, teniendo en cuenta el perfil de a quien se realiza el préstamo (si es profesor/investigador o estudiante).

		El Sistema debe confirmar una vez se ha registrado el préstamo, presentando un resumen del mismo (el cual podrá ser verificado por quien realiza el préstamo).
		El sistema debe ofrecer una alternativa para registrar el préstamo de equipos que por su bajo costo no tienen identificador alguno.
Consulta ficha de equipo	Como Laboratorista Quiero poder consultar la ficha de un determinado equipo Para poder tomar decisiones sobre equipos en los que se presenten novedades.	La ficha generada por el sistema debe ser equivalente a la manejada actualmente en excel (adjunto).
		Para consultar la ficha del equipo el laboratorista sólo debe ingresar su placa.
		La ficha del equipo debe salir en un formato que permita su impresión.
Realizar devolución global	Como laboratorista Quiero poder registrar la devolución de todos los ítems prestados a un cliente del laboratorio en un solo paso Para reducir el tiempo que toma dicho proceso	Al ingresar a la opción de devolución global, el sistema solicita el identificador de quien realizó el préstamo, y posteriormente muestra qué equipos tiene pendientes por devolverl.
		El laboratorista, al verificar que coincidan los ítems del préstamo, selecciona la opción de registrar devolución. Al hacer esto, todos los ítems asociados al préstamo deben cargarse nuevamente al inventario de productos disponibles.

1.4 Alcance

Permitir llevar tanto el control del uso adecuado de los equipos, como el aplicar más fácilmente las políticas de renovación y actualización de los mismos.

1.5 Supuestos

- Una persona puede tener más de un rol dentro de la ECI.

- Las imágenes que están asociadas al modelo y a las herramientas nunca van a ser borradas de la web.

1.6 Dependencias

DEPENDENCIA	OPERACIÓN
hibernate-jpa-2.0-api	Soluciona un error de persistencia relacionado a classNotFound
all-themes	Contiene todos los temas gratis que ofrece primefaces
shiro-core	Seguridad de la aplicación
shiro-web	Seguridad de la aplicación
shiro-ehcache	Seguridad de la aplicación
pax-swissbox-optional-jcl	Conjunto de utilidades de orientación OSGi para usar maven
jsf-api	framework de java para desarrollo de páginas web
jsf-impl	framework de java para desarrollo de páginas web
primefaces	librería de componentes para JavaServer Faces (JSF) de código abierto
junit	Conjunto de bibliotecas usadas para realizar pruebas unitarias
hamcrest-core	conjunto implementaciones matcher para operaciones comunes
guava	para hacer una busqueda usando google
javaee-web-api	Especificacion de apis para web
mysql-connector-java	Conectar la base de datos en mysql con java
log4j	permite a los desarrolladores de software escribir mensajes de registro
apache-log4j-extras	complementos para los mensajes de registro

h2	Crear una base de datos local para pruebas
mybatis	framework para manejar la persistencia de la base de datos en sql

1.7 Metodología de Prueba

El proceso de pruebas se realizó a medida que se iba sacando un método nuevo que se consideraba que era necesario probar (por ambigüedad de este o por alta dificultad de este), este conjunto de pruebas se encuentra en 3 clases ubicadas dentro del proyecto en "PdswProject/src/test", cada prueba es la representante de su determinada clase de equivalencia, un par de ejemplos de estas pruebas son:

- `cargar un modelo por nombre que contiene equipos asociados`

Es una prueba útil porque para la parte visual de la historia de uso "registrar equipo en el inventario" es necesario cargar los modelos inicialmente para que el usuario vea si ya existe o si hay que agregarlo y crear equipos de estos.

- `Cargar un usuario con préstamos de equipos normales, con préstamos de equipos básicos y con todos los roles`

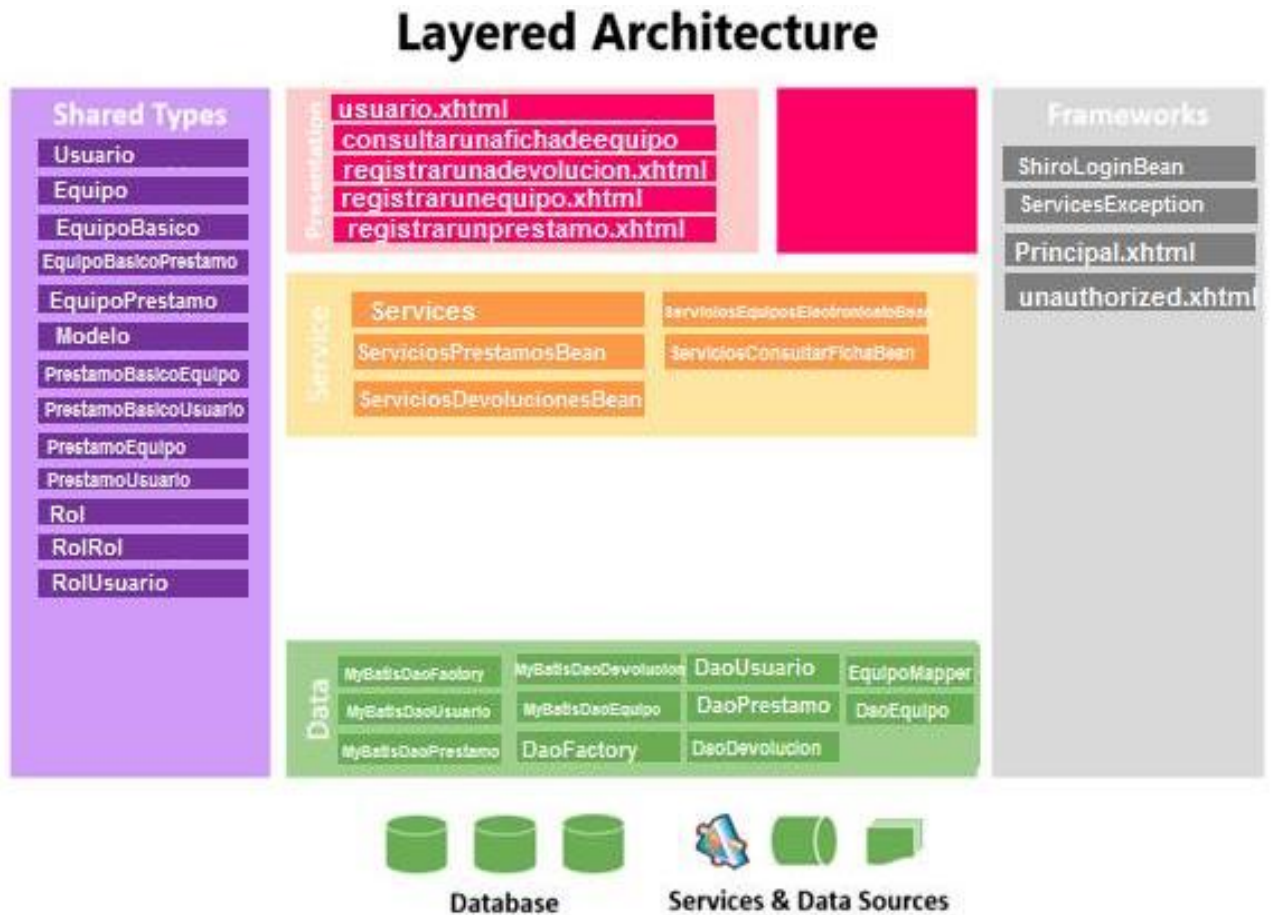
Es una prueba necesaria en varios casos de uso pero esta se hizo primordialmente para la historia de uso "registrar devolución", ya que cuando se carga un usuario, este tiene asociado una lista de sus préstamos tanto de equipos como de herramientas

1.8 Estándares seguidos

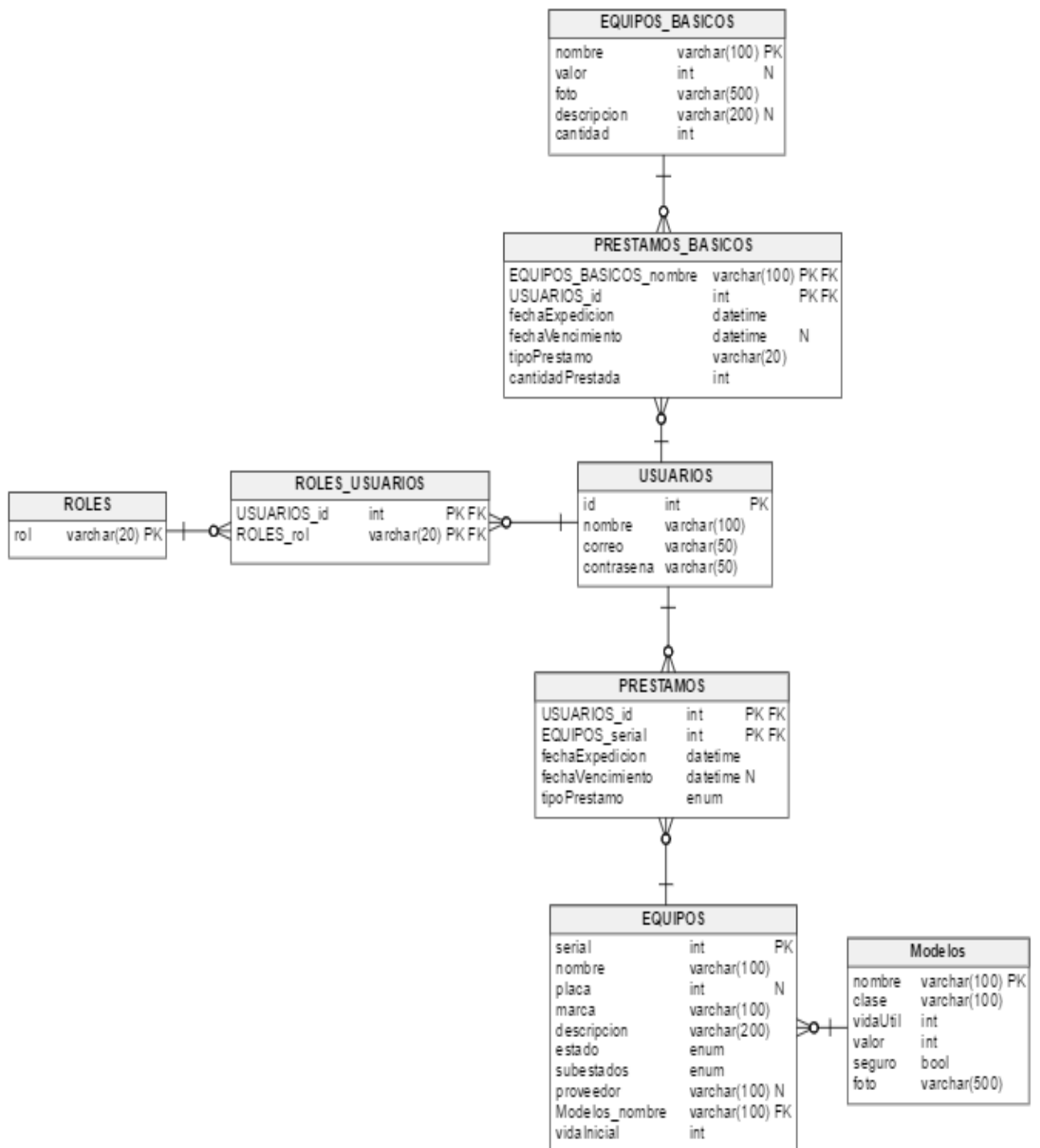
- Estándares de java
- Estándares UML

2 Diseño Detallado

2.1 Arquitectura de alto nivel



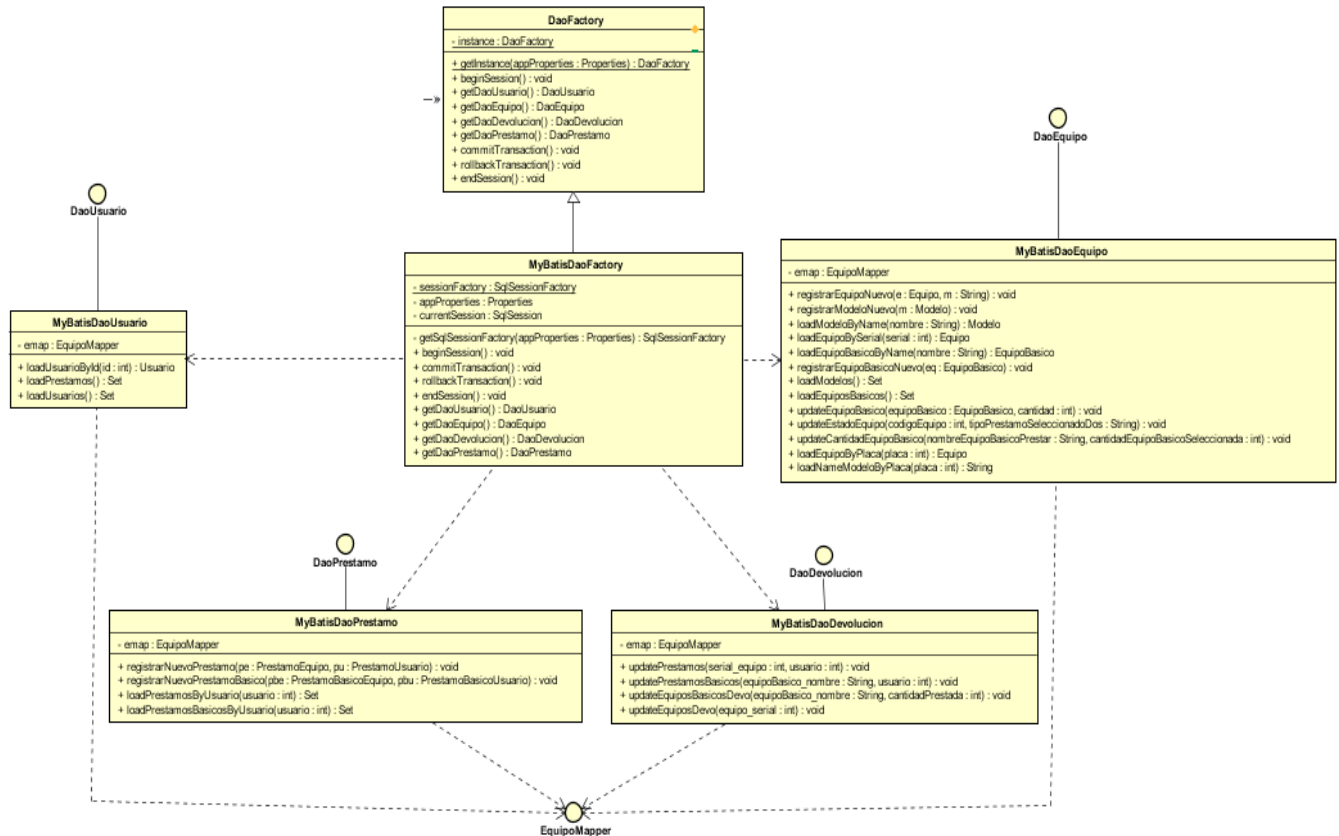
2.2 Modelo relacional



2.3 Diseño por paquetes/capas

2.3.1 Capa de persistencia

- Diagrama de clases

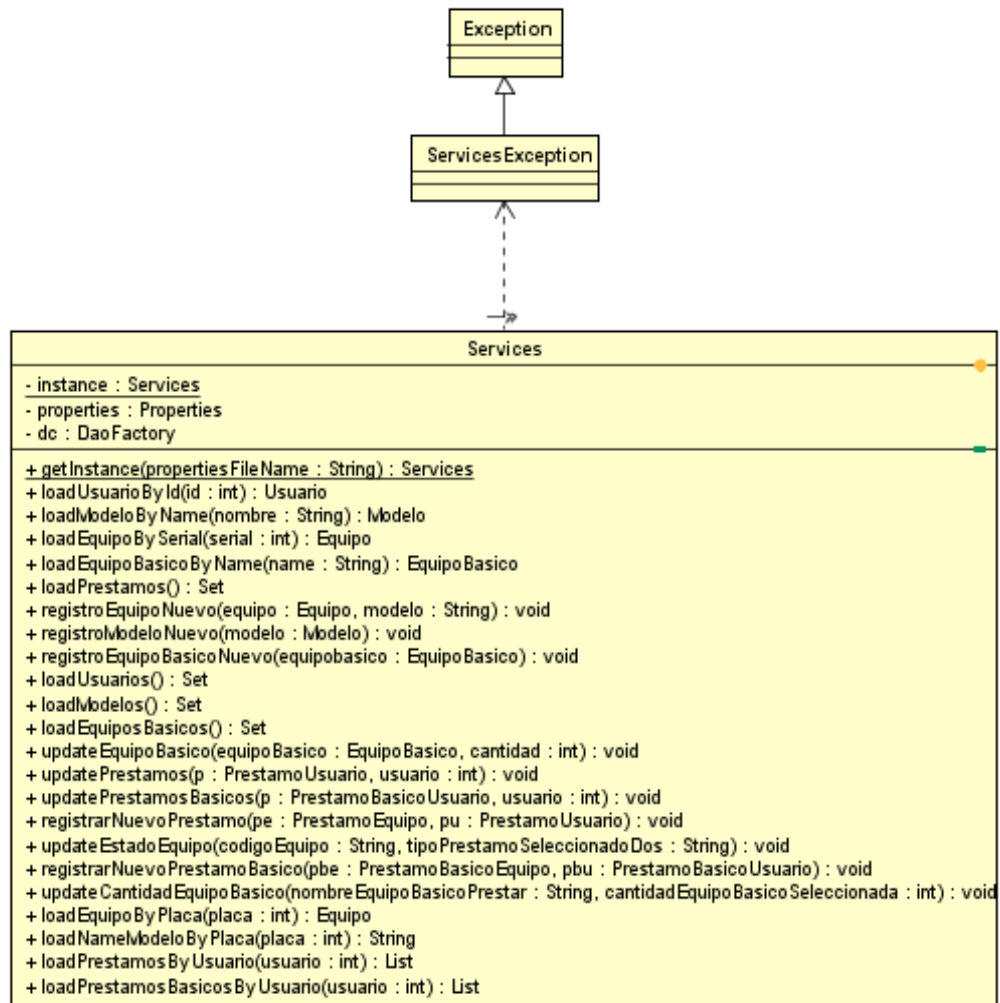


- Una descripción general de lo más relevante

La capa de persistencia consiste en los DAO (data-access-object) los cuales son extendidas por otras clases “MyBatisDao...” que son encargadas de usar el mapper para realizar las operaciones sobre la BD

2.3.2 Capa lógica

- Diagrama de clases



Usa entidades:

Usuario
<ul style="list-style-type: none"> - id : int - nombre : String - correo : String - contraseña : String - prestamos : Set - prestamosBasicos : Set - roles : Set
<ul style="list-style-type: none"> + getId() : int + setId(id : int) : void + getNombre() : String + setNombre(nombre : String) : void + getCorreo() : String + setCorreo(correo : String) : void + getContraseña() : String + setContraseña(contraseña : String) : void + getPrestamos() : Set + setPrestamos(prestamos : Set) : void + getPrestamosBasicos() : Set + setPrestamosBasicos(prestamosBasicos : Set) : void + getRoles() : Set + setRoles(roles : Set) : void + setPrestamosBasicos(prestamosBasicos : Set) : void + toString() : String

Modelo
<ul style="list-style-type: none"> - nombre : String - clase : String - vidaUtilHoras : int - valorComercial : int - seguro : Boolean - foto : String - equipos : Set
<ul style="list-style-type: none"> + getNombre() : String + setNombre(nombre : String) : void + getClase() : String + setClase(clase : String) : void + getVidaUtilHoras() : int + setVidaUtilHoras(vidaUtilHoras : int) : void + getValorComercial() : int + setValorComercial(valorComercial : int) : void + getSeguro() : Boolean + setSeguro(seguro : Boolean) : void + getFoto() : String + setFoto(foto : String) : void + getEquipos() : Set + setEquipos(equipos : Set) : void + toString() : String

Equipo
<ul style="list-style-type: none"> - serial : int - nombre : String - placa : int - vidaInicial : int - marca : String - descripcion : String - estado : String - subEstado : String - proveedor : String - prestamos : Set
<ul style="list-style-type: none"> + getSerial() : int + setSerial(serial : int) : void + getNombre() : String + setNombre(nombre : String) : void + getPlaca() : int + setPlaca(placa : int) : void + getMarca() : String + setMarca(marca : String) : void + getDescripcion() : String + setDescripcion(descripcion : String) : void + getEstado() : String + setEstado(estado : String) : void + getSubEstado() : String + setSubEstado(subEstado : String) : void + getProveedor() : String + setProveedor(proveedor : String) : void + getPrestamos() : Set + setPrestamos(prestamos : Set) : void + toString() : String + getVidaInicial() : int + setVidaInicial(vidaInicial : int) : void

EquipoPrestamo
<ul style="list-style-type: none"> - tipoPrestamo : String - equipoBasico : Equipo
<ul style="list-style-type: none"> + getTipoPrestamo() : String + setTipoPrestamo(tipoPrestamo : String) : void + getEquipoBasico() : Equipo + setEquipoBasico(equipoBasico : Equipo) : void

EquipoBasico
<ul style="list-style-type: none"> - nombre : String - valor : int - foto : String - descripcion : String - cantidadInventario : int - prestamosBasicos : Set
<ul style="list-style-type: none"> + getNombre() : String + setNombre(nombre : String) : void + getValor() : int + setValor(valor : int) : void + getFoto() : String + setFoto(foto : String) : void + getDescripcion() : String + setDescripcion(descripcion : String) : void + getCantidadInventario() : int + setCantidadInventario(cantidadInventario : int) : void + getPrestamosBasicos() : Set + setPrestamosBasicos(prestamosBasicos : Set) : void + toString() : String

EquipoBasicoPrestamo
<ul style="list-style-type: none"> - equipoBasico : EquipoBasico - cantidad : int - tipoPrestamo : String
<ul style="list-style-type: none"> + getEquipoBasico() : EquipoBasico + setEquipoBasico(equipoBasico : EquipoBasico) : void + getCantidad() : int + setCantidad(cantidad : int) : void + getTipoPrestamo() : String + setTipoPrestamo(tipoPrestamo : String) : void

PrestamoBasicoUsuario
<ul style="list-style-type: none"> - equipoBasico_nombre : String - fechaExpedicion : Date - fechaVencimiento : Date - tipoPrestamo : String - cantidadPrestada : int
<ul style="list-style-type: none"> + getEquipoBasico_nombre() : String + setEquipoBasico_nombre(equipoBasico_nombre : String) : void + getFechaExpedicion() : Date + setFechaExpedicion(fechaExpedicion : Date) : void + getFechaVencimiento() : Date + setFechaVencimiento(fechaVencimiento : Date) : void + getTipoPrestamo() : String + setTipoPrestamo(tipoPrestamo : String) : void + getCantidadPrestada() : int + setCantidadPrestada(cantidadPrestada : int) : void + toString() : String

PrestamoBasicoEquipo
<ul style="list-style-type: none"> - usuario_id : int - fechaExpedicion : Date - fechaVencimiento : Date - tipoPrestamo : String - cantidadPrestada : int
<ul style="list-style-type: none"> + getUsuario_id() : int + setUsuario_id(usuario_id : int) : void + getFechaExpedicion() : Date + setFechaExpedicion(fechaExpedicion : Date) : void + getFechaVencimiento() : Date + setFechaVencimiento(fechaVencimiento : Date) : void + getTipoPrestamo() : String + setTipoPrestamo(tipoPrestamo : String) : void + getCantidadPrestada() : int + setCantidadPrestada(cantidadPrestada : int) : void + toString() : String

PrestamoUsuario
<ul style="list-style-type: none"> - equipo_serial : int - fechaExpedicion : Date - fechaVencimiento : Date - tipoPrestamo : String
<ul style="list-style-type: none"> + getEquipo_serial() : int + setEquipo_serial(equipo_serial : int) : void + getFechaExpedicion() : Date + setFechaExpedicion(fechaExpedicion : Date) : void + getFechaVencimiento() : Date + setFechaVencimiento(fechaVencimiento : Date) : void + getTipoPrestamo() : String + setTipoPrestamo(tipoPrestamo : String) : void + toString() : String

PrestamoEquipo
<ul style="list-style-type: none"> - usuario_id : int - fechaExpedicion : Date - fechaVencimiento : Date - fechaExpedicionString : String - fechaVencimientoString : String - tipoPrestamo : String
<ul style="list-style-type: none"> + getUsuario_id() : int + setUsuario_id(usuario_id : int) : void + getFechaExpedicion() : Date + setFechaExpedicion(fechaExpedicion : Date) : void + getFechaVencimiento() : Date + setFechaVencimiento(fechaVencimiento : Date) : void + getTipoPrestamo() : String + setTipoPrestamo(tipoPrestamo : String) : void + toString() : String + getFechaExpedicionString() : String + setFechaExpedicionString(fechaExpedicionString : String) : void + getFechaVencimientoString() : String + setFechaVencimientoString(fechaVencimientoString : String) : void

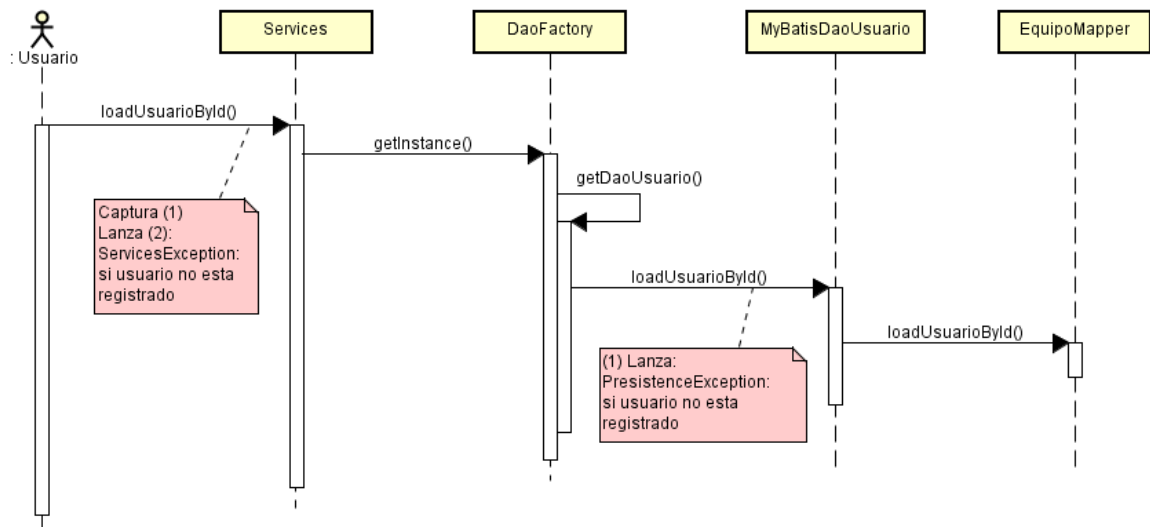
Rol
<ul style="list-style-type: none"> - rol : String - rolesdeusuarios : Set
<ul style="list-style-type: none"> + getRol() : String + setRol(rol : String) : void + getRolesdeusuarios() : Set + setRolesdeusuarios(rolesdeusuarios : Set) : void + toString() : String

RolRol
<ul style="list-style-type: none"> - usuario_id : int
<ul style="list-style-type: none"> + getUsuario_id() : int + setUsuario_id(usuario_id : int) : void + toString() : String

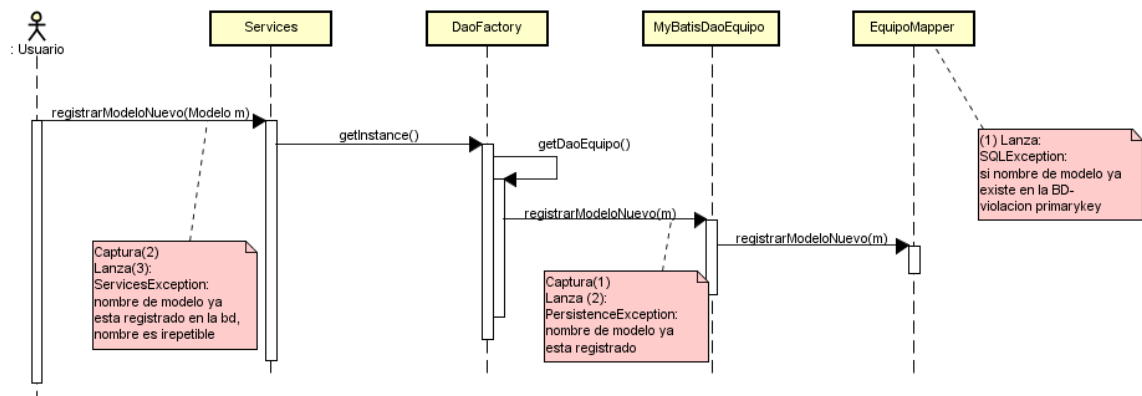
RolUsuario
<ul style="list-style-type: none"> - rol_r : String
<ul style="list-style-type: none"> + getRol_r() : String + setRol_r(rol_r : String) : void + toString() : String

- *Dos diagramas de secuencia*

- Cargar un usuario dado su identificador



- Registrar un nuevo modelo en la BD, dada la entidad modelo



- *Una descripción con lo más relevante*

En la lógica la estructura siempre es muy similar para cada método ofrecido en los servicios, me refiero a el camino que toma hasta llega al xml donde están mapeados los métodos, lo primero es instanciar el daofactory, luego este “getea” al dao correspondiente (`MyBatisDaoEquipo`, `MyBatisDaoUsuario`, `MyBatisDaoPrestamo` etc...) , en estas clases lo que se hace es delegar este método a la interfaz que está conectada al mapper (`EquipoMapper`) y allí se hace la SQLquery correspondientes y los resultMap posibles.

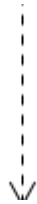
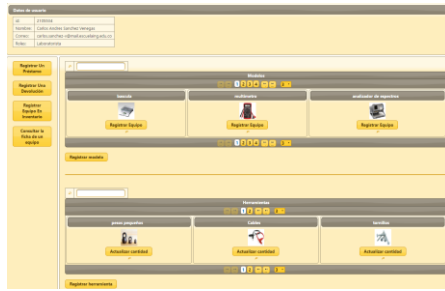
2.3.3 Capa de presentación

- ShiroLoginBean



ShiroLoginBean
<ul style="list-style-type: none">- log : Logger- username : String- password : String- rememberMe : Boolean- botonRegistrarEquipoEnInventario : boolean- botonRegistrarUnaDevolucion : boolean- botonRegistrarUnPrestamo : boolean- botonConsultarFicha : boolean
<ul style="list-style-type: none">+ getSubject() : Subject+ doLogin() : void- facesError(message : String) : void+ getUsername() : String+ setUsername(login : String) : void+ getPassword() : String+ setPassword(senha : String) : void+ getRememberMe() : Boolean+ setRememberMe(lembrar : Boolean) : void+ isBotonRegistrarEquipoEnInventario() : boolean+ setBotonRegistrarEquipoEnInventario(botonRegistrarEquipoEnInventario : boolean) : void+ isBotonRegistrarUnaDevolucion() : boolean+ setBotonRegistrarUnaDevolucion(botonRegistrarUnaDevolucion : boolean) : void+ isBotonRegistrarUnPrestamo() : boolean+ setBotonRegistrarUnPrestamo(botonRegistrarUnPrestamo : boolean) : void+ isBotonConsultarFicha() : boolean+ setBotonConsultarFicha(botonConsultarFicha : boolean) : void

- ServicioEquiposElectronicatobean



- ServiciosConsultarFichaBean

Datos de usuario	
id:	2105534
Nombre:	Carlos Andres Sanchez Venegas
Correo:	carlos.sanchez-v@mail.escuelaing.edu.co
Roles:	Laboratorista

<div>Registrar Un Préstamo</div> <div>Registrar Una Devolución</div> <div>Registrar Equipo En Inventario</div> <div>Consultar la ficha de un equipo</div>	<div>Equipo a consultar</div> <div>Placa: <input type="text"/></div> <div>Enviar</div>
---	--



ServiciosConsultarFichaBean
- loginBean : ShiroLoginBean - services : Services - placaAconsultar : String - equiposeleccionado : Equipo - modeloAsociado : Modelo - equipoexiste : Boolean
+ limpiarPaginaConsultarFicha() : void + demeloTodo() : String + accionConsultarFicha() : void + getLoginBean() : ShiroLoginBean + setLoginBean(loginBean : ShiroLoginBean) : void + getServices() : Services + setServices(services : Services) : void + getPlacaAconsultar() : String + setPlacaAconsultar(placaAconsultar : String) : void + getEquiposeleccionado() : Equipo + setEquiposeleccionado(equiposeleccionado : Equipo) : void + getModeloAsociado() : Modelo + setModeloAsociado(modeloAsociado : Modelo) : void + getEquipoexiste() : Boolean + setEquipoexiste(equipoexiste : Boolean) : void

- ServiciosDevolucionesBean

ServiciosDevolucionesBean
<ul style="list-style-type: none"> - loginBean : ShiroLoginBean - yaBusqueEquipoADevolver : boolean - serialDevolucionEncontrado : boolean - serialDevolucionNoEncontrado : boolean - textoSalidaEquipoADevolver : String - serialADevolver : String - usuarioDevolucion : Usuario - horaActual : Date - condigoEstudianteBasicos : String - nombreEquipoBasicoDevolver : String - cantidadBasicaDevuelta : int - usuarioDevolucionBasico : Usuario - codigoUsuario : String - usuarioExiste : boolean - listaPrestamoActuales : List - listaPrestamoBasicoActuales : List - nombresEquiposBasicos : HashMap - se : Services
<ul style="list-style-type: none"> + botonAceptarDevolucionGlobal() : void + botonFinalDevolucionGlobal() : void + limpiarRegistrarDevolucionGlobal() : void + limpiarDevolucion() : void + limpiarDevolucionBasica() : void + limpiarPaginaRegistrarUnaDevolucion() : void + accionBuscarDevolucion() : void + accionRealizarDevolucion() : void + accionRealizarDevolucionBasica() : void + isYaBusqueEquipoADevolver() : boolean + setYaBusqueEquipoADevolver(yaBusqueEquipoADevolver : boolean) : void + isSerialDevolucionEncontrado() : boolean + setSerialDevolucionEncontrado(serialDevolucionEncontrado : boolean) : void + isSerialDevolucionNoEncontrado() : boolean + setSerialDevolucionNoEncontrado(serialDevolucionNoEncontrado : boolean) : void + getTextoSalidaEquipoADevolver() : String + setTextoSalidaEquipoADevolver(textoSalidaEquipoADevolver : String) : void + getSerialADevolver() : String + setSerialADevolver(serialADevolver : String) : void + getUsuarioDevolucion() : Usuario + setUsuarioDevolucion(usuarioDevolucion : Usuario) : void + getCondigoEstudianteBasicos() : String + setCondigoEstudianteBasicos(condigoEstudianteBasicos : String) : void + getNombreEquipoBasicoDevolver() : String + setNombreEquipoBasicoDevolver(nombreEquipoBasicoDevolver : String) : void + getCantidadBasicaDevuelta() : int + setCantidadBasicaDevuelta(cantidadBasicaDevuelta : int) : void + getUsuarioDevolucionBasico() : Usuario + setUsuarioDevolucionBasico(usuarioDevolucionBasico : Usuario) : void + getLoginBean() : ShiroLoginBean + setLoginBean(loginBean : ShiroLoginBean) : void + getNombresEquiposBasicos() : HashMap + setNombresEquiposBasicos(nombresEquiposBasicos : HashMap) : void + getCodigoUsuario() : String + setCodigoUsuario(codigoUsuario : String) : void + isUsuarioExiste() : boolean + setUsuarioExiste(usuarioExiste : boolean) : void + getListPrestamoActuales() : List + setListaPrestamoActuales(listaPrestamoActuales : List) : void + getListPrestamoBasicoActuales() : List + setListaPrestamoBasicoActuales(listaPrestamoBasicoActuales : List) : void

- ServiciosPrestamosBean

Datos de usuario

id: 210554

nombre: Carlos Andres Sanchez Vargas

correo: carlos.sanchez@gmail.com

roles: Laboratorio

Registrar Un Prestamo

Registrar Una Devolucion

Registrar Equipo En Inventario

Consultar la ficha de un equipo

Iniciar

Ingresar codigo de usuario: 210545

OK

Datos usuario que pide el prestamo

Nombre: Jhony Edwin Salas

Correo: jhony.salas@gmail.com

Roles: estudiante, laboratorio, profesor

Agregar equipo a prestamo

Seleccione tipo de equip:

Lista Equipos a prestar

Lista equipos

No records found.

Lista incrementos

No records found.

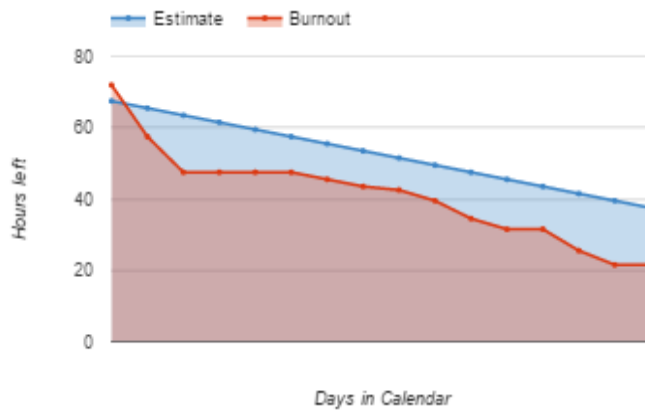
Completar prestamo

ServiciosPrestamosBean
<ul style="list-style-type: none"> - services : Services - loginBean : ShiroLoginBean - codigoUsuarioPrestamo : String - estudiante Existe : boolean - usuario Seleccionado : Usuario - rolesUsuario Seleccionado : String - nombreEquipoBasico Prestar : String - codigoEquipo : String - tipoEquipo : String - codificado : boolean - no Codificado : boolean - tipoPrestamo Seleccionado : String - nombresTipo Prestamo : List - listaEquipos : List - listaEquipos Basicos : List - equipo Seleccionado : EquipoPrestamo - prestamoEquipoBasico Seleccionado : EquipoBasicoPrestamo - cantidadEquipoBasico Seleccionada : int - tipoPrestamo Seleccionado Dos : String
<pre> + limpiarPaginaRegistrarUnPrestamo() : void + accionCompletarPrestamo() : void + accionRegistrarPrestamoEquipo() : void + accionRegistrarPrestamoEquipoBasico() : void + accionCambiarTipo() : void + accionBotonUsuarioPrestamo() : void + setCodigoUsuarioPrestamo(codigo : String) : void + getCodigoUsuarioPrestamo() : String + getLoginBean() : ShiroLoginBean + setLoginBean(loginBean : ShiroLoginBean) : void + isEstudianteExiste() : boolean + setEstudianteExiste(estudianteExiste : boolean) : void + getUsuarioSeleccionado() : Usuario + setUsuarioSeleccionado(usuarioSeleccionado : Usuario) : void + getRolesUsuarioSeleccionado() : String + setRolesUsuarioSeleccionado(rolesUsuarioSeleccionado : String) : void + getNombreEquipoBasicoPrestar() : String + setNombreEquipoBasicoPrestar(nombreEquipoBasicoPrestar : String) : void + getCodigoEquipo() : String + setCodigoEquipo(codigoEquipo : String) : void + getTipoEquipo() : String + setTipoEquipo(tipoEquipo : String) : void + isCodificado() : boolean + setCodificado(codificado : boolean) : void + isNoCodificado() : boolean + setNoCodificado(noCodificado : boolean) : void + getTipoPrestamoSeleccionado() : String + setTipoPrestamoSeleccionado(tipoPrestamoSeleccionado : String) : void + getNombresTipoPrestamo() : List + setNombresTipoPrestamo(nombresTipoPrestamo : List) : void + getListEquipos() : List + setListaEquipos(listaEquipos : List) : void + getListaEquiposBasicos() : List + setListaEquiposBasicos(listaEquiposBasicos : List) : void + getEquipoSeleccionado() : EquipoPrestamo + setEquipoSeleccionado(equipoSeleccionado : EquipoPrestamo) : void + getCantidadEquipoBasicoSeleccionada() : int + setCantidadEquipoBasicoSeleccionada(cantidadEquipoBasicoSeleccionada : int) : void + getPrestamoEquipoBasicoSeleccionado() : EquipoBasicoPrestamo + setPrestamoEquipoBasicoSeleccionado(prestamoEquipoBasicoSeleccionado : EquipoBasicoPrestamo) : void + getTipoPrestamoSeleccionadoDos() : String + setTipoPrestamoSeleccionadoDos(tipoPrestamoSeleccionadoDos : String) : void </pre>

- Sprint 2:
 - Backlog

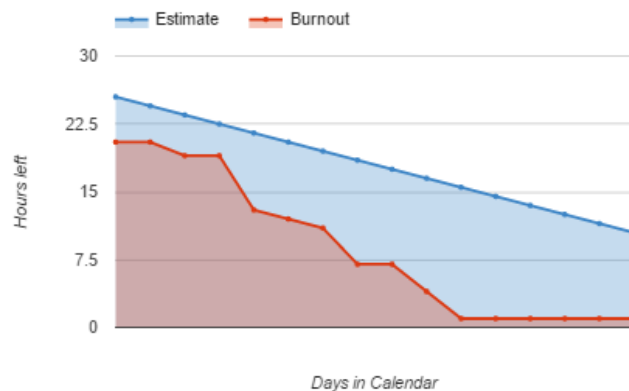
User story	Task	Developer	Time (estimate)	Time (spent)	Time (left)	4/27/2016	4/28/2016	4/29/2016	4/30/2016	5/1/2016	5/2/2016	5/3/2016	5/4/2016	5/5/2016	5/6/2016	5/7/2016	5/8/2016	5/9/2016	5/10/2016	5/11/2016
Registrar equipo en el inventario	Mejorar las vistas	Carlos Sanchez	15	15	7	8														
Registrar equipo en el inventario	Mejorar las vistas	Jhordy Salinas	10	10	3.5	6	0.5													
Registrar equipo en el inventario	títulos, ortografía, formato tabla, etc.	Jhordy Salinas	6	6	3.5		2.5													
Registrar equipo en el inventario	Seleccionar estado del equipo en lugar de digitarlo	Jhordy Salinas	3	3	2.5	0.5														
Registrar equipo en el inventario	Facilidad de búsqueda aproximada (por ejemplo autocompletar)	Carlos Sanchez	5	5	2		3													
Registrar devolucion	Terminar lo faltante del sprint anterior	Leonardo Herrera	5	8	0		4			2	2									
Registrar Prestamo	Crear vista registrar prestamo	Carlos Sanchez	10	11	0								1	5					5	
Registrar Prestamo	Conectar vista con logico registrar prestamo	Carlos Sanchez	6	6	2											3			1	
Registrar Prestamo	separar manageBean	Jhordy Salinas	1	1	0								1							
Registrar Prestamo	Parte logica registrar prestamo	Jhordy Salinas	3	3	0														3	
Registrar Prestamo	Crear vista registrar prestamo	Jhordy Salinas	3	3	1									2						
Registrar devolucion	Terminar lo faltante del sprint anterior	Jhordy Salinas	0.5	1	0														1	
			0	0	0															
			0	0	0															
			0	0	0															
			0	0	0															
			0	0	0															
TOTAL			67.5	72	21.5	14.5	10	0	0	0	2	2	1	3	5	3	0	6	4	0
Daily burnout				37.5		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Total time left (from estimate)					67.5	65.5	63.5	61.5	59.5	57.5	55.5	53.5	51.5	49.5	47.5	45.5	43.5	41.5	39.5	37.5
Total time left (from spent)					72	57.5	47.5	47.5	47.5	47.5	45.5	43.5	42.5	39.5	34.5	31.5	31.5	25.5	21.5	21.5

- Burndown-chart

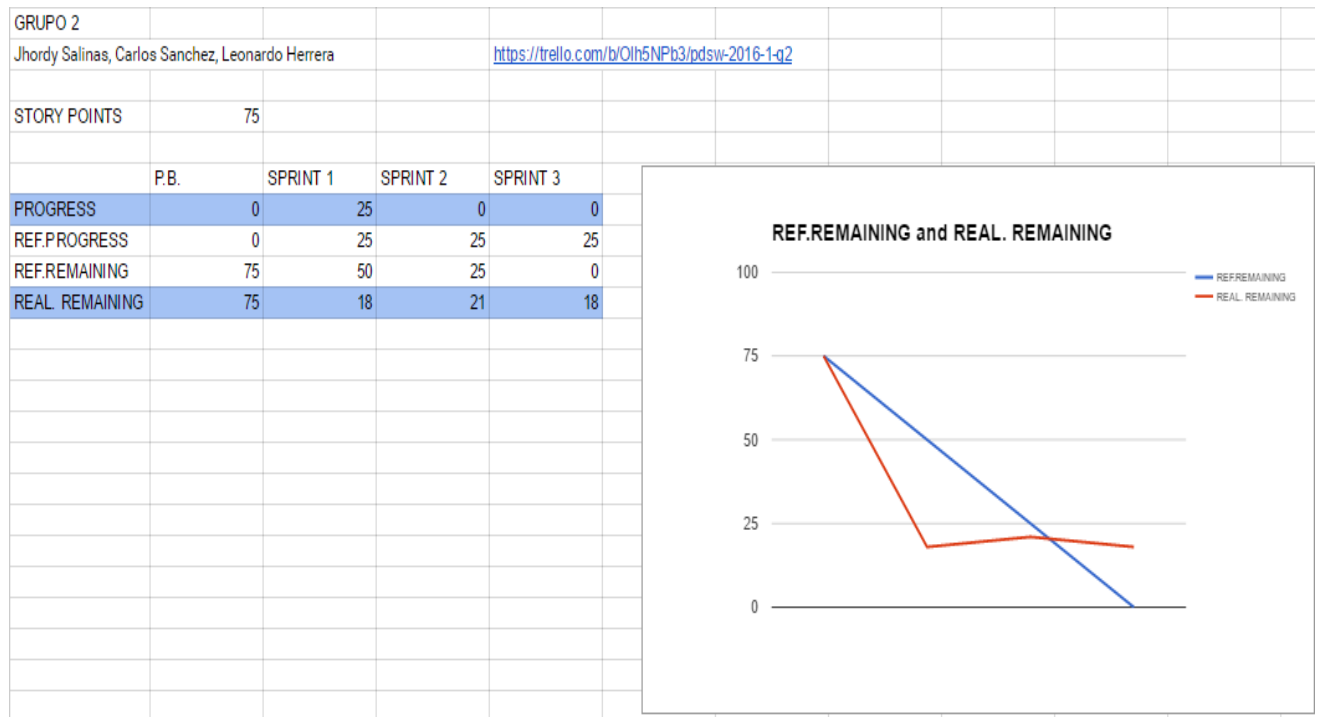


User story	Task	Developer	Time (estma)	Time (spent)	Time (left)	S0006	S1006	S2006	S3006	S4006	S5006	S6006	S7006	S8006	S9006	S0007	S1007	S2007	S3007	S4007	S5007	S6007
Registrar equipo en inventario	Resolver sugerencia de el dueño de producto y stakeholder de cambiar nombre "equipo basico" a "herramienta"	Carlos Sanchez	0.5	0.5	0				0.5													
Registrar equipo en inventario	Resolver sugerencia de el dueño de producto y stakeholder de cambiar nombre "crear equipo" a "registrar equipo"		0.5	0.5	0				0.5													
Registrar equipo en inventario	Resolver sugerencia de el dueño de producto y stakeholder de limpiar formularios	Carlos Sanchez	1	1	0				1													
Registrar devolucion	Realizar devolución global	Carlos Sanchez	6	3	3				3													
Registrar equipo en inventario	Resolver sugerencia de el dueño de producto y stakeholder de dejar como default cada vez que cree un equipo que este quede como estado "Activo" y subestado "En almacén"	Jhordy Salinas	0.5	0.5	0		0.5															
Registrar equipo en inventario	Resolver sugerencia de el dueño de producto y stakeholder de ordenar los botones por jerarquia	Jhordy Salinas	0.5	0.5	0		0.5															
Registrar devolucion	arreglar exoeccion de campo vacio en serial devolucion	Jhordy Salinas	1	1	0.5		0.5															
Consulta ficha de equipo	todo lo qe se necesita para consultar ficha	Jhordy Salinas	8	6	2					1	1	4										
Registrar equipo en inventario	Resolver sugerencia de el dueño de producto y stakeholder de poner de ultimas serial que son datos que al ponerlos se acaba	Leonardo Herrera	0.5	0.5	0				0.5													
Registrar equipo en inventario	Resolver sugerencia de el dueño de producto y stakeholder de aparte de tener vida util por equipo tambien agregar atributo de equipo vida util inicial	Leonardo Herrera	1	1	0.5				0.5													
Registrar equipo en inventario	Resolver sugerencia de el dueño de producto y stakeholder de poder agregar el manual del equipo como archivo PDF	Leonardo Herrera	3	3	0									1	2							
Registrar un prestamo	Resolver sugerencia de el dueño de producto y stakeholder de definir todo el conjunto de cosas como un solo prestamo, osea como una bolsa que va agregando	Leonardo Herrera	3	3	0									2	1							
					0																	
					0																	
TOTAL			25.5	20.5	6	0	1.5	0	6	1	1	4	0	3	3	0	0	0	0	0	0	0
Daily burnout			10.5			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Total time left (from estimate)				25.5		24.5	23.5	22.5	21.5	20.5	19.5	18.5	17.5	16.5	15.5	14.5	13.5	12.5	11.5	10.5		
Total time left (from spent)				20.5		20.5	19	19	13	12	11	7	7	4	1	1	1	1	1	1		

- *Burndown*-chart



3.3 Release-Burndown-chart



3.4 Métricas por desarrollador

Los 3 desarrolladores trabajaron desarrollando las historias de uso planteadas en Trello, el trabajo para cada Sprint se planteaba para que fuera equilibrado y las métricas de cada desarrollador son iguales para todos, me refiero a que el trabajo desempeñado por cada uno es muy homogéneo al del resto del equipo.

Glosario (Definiciones y Siglas)

Scrum: proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto

MDD: “Desarrollo dirigido por modelos”, es una metodología de desarrollo de software que se centra en la creación y explotación de modelos de dominio (es decir, representaciones abstractas de los conocimientos y actividades que rigen un dominio de aplicación particular), más que en conceptos informáticos (o algoritmos).

TDD: “Desarrollo dirigido por pruebas”, el propósito del desarrollo guiado por pruebas es lograr un código limpio que funcione. La idea es que los requisitos sean traducidos a pruebas, de este modo, cuando las pruebas pasen se garantizará que el software cumple con los requisitos que se han establecido.

BDD: “Desarrollo dirigido por comportamiento”, BDD busca un lenguaje común para unir la parte técnica y la de negocio, y que sea desde ese lenguaje común desde donde arranque el Testing y, desde ahí, el desarrollo.

XP: La programación extrema es una metodología de desarrollo ágil que tiene como principal objetivo aumentar la productividad a la hora de desarrollar un proyecto software. Da prioridad a los trabajos que dan un resultado directo y en los cuales se reduce la burocracia que pueda existir en el entorno de trabajo.

Backlog: Artefacto (documento) de Scrum

(release) Burndown-chart: Artefacto (gráfica) de Scrum

Sprint: El período en el cual se lleva a cabo el trabajo de las tareas en sí (1 a 4 semanas).

UML: es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

ECI: Escuela Colombiana de Ingeniería

Mutex: se usan en programación concurrente para evitar el ingreso a sus secciones críticas por más de un proceso a la vez.