

Equipo “Panda-LC”

Mi Dientito :)

Software Architecture Document

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Equipo “Panda-LC”

Revision History

Date	Version	Description	Author
4/mar/2016	<1.0>	Etapa inicial del proyecto final	Melissa Jazmín López Soto Mariana Itzel Velasco Cortés María Fernanda Vela Huerta

Equipo “Panda-LC”

Table of Contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, Acronyms, and Abbreviations
 - 1.4 References
 - 1.5 Overview
2. Architectural Representation
3. Architectural Goals and Constraints
4. Use-Case View
5. Logical View
 - 5.1 Overview
 - 5.2 Architecturally Significant Design Packages
 - 5.3 Use-Case Realizations
6. Process View
7. Deployment View
8. Implementation View
 - 8.1 Overview
 - 8.2 Layers
9. Data View (optional)
10. Size and Performance

Equipo “Panda-LC”

11. Quality

Equipo “Panda-LC”

Software Architecture Document

1. Introduction

Actualmente vivimos en una sociedad que es totalmente activa, donde el tiempo es muy valioso y muchas veces las citas médicas se convierten en un proceso tedioso, además de que muchas veces olvidamos los medicamentos o procedimientos médicos que debemos hacer a lo largo del día. Mi Dientito surge con el propósito de mejorar la relación entre médico-paciente, permitiendo a los usuarios tener un mejor control de sus citas y medicamentos.

Mi Dientito es una aplicación móvil que enlaza consultas entre dentistas y pacientes; es decir, los dentistas pueden dar de alta a sus pacientes y administrarlos, incluso mandar recetas médicas y sus próximas citas. En el caso de los pacientes, pueden ver sus recetas, historial, farmacias cercanas a su ubicación y recordatorios de sus próximas citas.

Purpose

El propósito de este documento es dar una visión general de la arquitectura del sistema, al usar distintas vistas arquitectónicas para representar los aspectos correspondientes del mismo. Se tiene la intención de captar y transmitir las decisiones arquitectónicas significativas que se hayan introducido en el sistema.

Scope

Este Documento de Arquitectura de Software (SAD) describe la arquitectura y diseño del sistema Mi Dientito; tiene como objetivo proporcionar al equipo encargado del desarrollo y soporte con información y referencias

Equipo “Panda-LC”

para permitir un apoyo efectivo.

Definitions, Acronyms, and Abbreviations

Término	Definición
UML	Unified Modelling Language es un lenguaje de modelado de sistemas de software. Para más información visite www.uml.org
Aplicación web	Herramienta a la que usuarios pueden acceder a través de un servidor web, haciendo uso de Internet o de una intranet mediante un navegador. Para más información visite http://rua.ua.es/dspace/bitstream/10045/4412/5/03c-AplicacionesWeb.pdf
Portabilidad	Característica que tiene un software para ejecutarse en diferentes plataformas. Para más información visite http://www.ecured.cu/Portabilidad
Bus	Conjunto de conexiones físicas que pueden compartirse con múltiples componentes de hardware para que se comuniquen entre sí. Para más información visite http://es.ccm.net/contents/364-que-es-un-bus-informatico
LAN	Grupo de equipos que pertenecen a la misma organización y están conectados dentro de un área

Equipo “Panda-LC”

	geográfica pequeña a través de una red. Para más información visite http://es.ccm.net/contents/253-lan-red-de-area-local
Fiabilidad	Característica de los sistemas informáticos por la que se mide el tiempo de funcionamiento sin fallos. Para más información visite http://www.mastermagazine.info/termino/4990.php
Extensibilidad	La extensibilidad es un requisito no funcional y un principio de diseño de sistemas en el que la implementación tiene en cuenta la incorporación de nuevas características. Para más información visite http://botica-informatica.blogspot.mx/2014/01/extensibilidad.html

References

Los siguientes documentos y materiales de referencias deberán ser consultados simultáneamente con este escrito.

Documento/Referencia	Versión	Fecha	Producido por
Course Registration System. Software Architecture Document http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm	1.0	Mar, 1999	S. Johnson

Equipo “Panda-LC”

Unified Modeling Language http://www.uml.org/	N/A	Jan, 2016	Object Management Group

Overview

[This subsection describes what the rest of the **Software Architecture Document** contains and explains how the **Software Architecture Document** is organized.]

Sección	Descripción
2	Representación Arquitectónica En esta sección se describe cuál es la arquitectura de software del sistema actual y la forma en la que se representa. De los Casos de Uso, Logical, Deployment y las Vistas de Implementación, se enumeran los puntos que son necesarios y se explican los tipos de elementos de modelo que contiene.
3	Objetivos y Limitaciones Arquitectónicas En esta sección se describen los requisitos de software y los objetivos que tienen impacto en la arquitectura, por ejemplo, la seguridad, privacidad, y el uso del producto off-the-shelf, portabilidad, distribución y reutilización. También se

Equipo “Panda-LC”

	mencionan las limitaciones especiales que pueden aplicarse: diseño y aplicación de estrategias, herramientas de desarrollo, estructura del equipo, calendarización, etc.
4	Casos de Uso En esta sección se enumeran los casos de uso o escenarios a partir del modelo de casos de uso si representan algo significativo, funcionalidad central en el sistema final, o si tienen amplia cobertura de arquitectura.
5	Logical View En esta sección se muestran los componentes arquitectónicos significativos del sistema; entre ellos están las interfaces.
6	Process View En esta sección se describe la descomposición del sistema en procesos ligeros (hilos individuales de control) y los procesos pesados (agrupaciones de procesos ligeros). Se organizan las secciones por grupos de proceso que se comunican o interactúan; y se mencionan los principales modos de comunicación entre procesos, como el paso de mensajes, interrupciones y de encuentro.
7	Deployment View En esta sección se describe las configuraciones de red físicas (hardware) en las que se debe desplegar y ejecutar el software. Como mínimo para cada configuración se indican los nodos físicos (computadoras, CPU) que ejecutan el

Equipo “Panda-LC”

	software, y sus interconexiones (bus, LAN, point-to-point, etc).
8	Implementation View En esta sección se describe la estructura general del modelo de implementación, la descomposición del software en capas y subsistemas y los componentes arquitectónicos indispensables.
9	Data View En esta sección se describe la perspectiva del almacenamiento de datos persistente del sistema.
10	Size and Performance En esta sección se describen las características principales de dimensionamiento del software que tienen un impacto en la arquitectura, incluso las restricciones de rendimiento objetivo
11	Quality En esta sección se describe cómo la arquitectura de software contribuye a todas las capacidades del sistema: extensibilidad, fiabilidad, portabilidad, entre otras.

2. Architectural Representation

Este documento detalla la arquitectura usando el “RUP naming convention”. Las vistas que se utilizaron para documentar la aplicación son:

- Vista Lógica:
 - Audiencia: Diseñadores
 - Área: Requerimientos Funcionales -> se describe el diseño de

Equipo “Panda-LC”

modelo de objetos. Así mismo, se describen las realizaciones de los casos de uso más importantes.

- Relación con: Modelo de Diseño.
- Vista de Proceso:
 - Audiencia: Integradores
 - Área: Requerimientos No Funcionales -> se describe la concurrencia del diseño y los aspectos de sincronización.
 - Relación con: Ninguno en especial.
- Vista de Implementación:
 - Audiencia: Programadores
 - Área: Componentes de Software -> se describen las capas y subsistemas de la aplicación.
 - Relación con: Modelo de Implementación y componentes.
- Vista de Despliegue:
 - Audiencia: Administradores de despliegue.
 - Área: Topología -> se describe el mapeo del software en el hardware y muestra los aspectos del sistema distribuido.
 - Relación con: Modelo de despliegue.
- Vista de Casos de Uso:
 - Audiencia: Todos los que usarán el sistema, incluyendo los usuarios finales.
 - Área: Se describe el conjunto de escenarios y de los casos de uso que representan un aspecto significativo y que sea de funcionalidad central para el sistema
 - Relación con: Modelo de Casos de Uso, Documentos de Casos de Uso

Equipo “Panda-LC”

3. Architectural Goals and Constraints

A continuación se listan todos los requerimientos que han sido identificados para que el sistema se pueda desarrollar. El orden en el que se encuentran es el orden en el que se atenderán:

- I. Los usuarios del sistema deben tener un dispositivo con conexión a Internet para poder hacer uso del sistema.
- II. Para poder acceder al sistema, los usuarios deben estar registrados.
- III. Registrar dentista, un dentista puede registrarse por sí solo en el sistema, llenando un formulario con su información correspondiente.
- IV. Registrar a los pacientes del dentista correspondiente.
- V. Encriptar información personal de dentistas y pacientes.
- VI. Poder generar nuevas citas y especificar el motivo de éstas.
- VII. Desplegar citas del dentista.
- VIII. Desplegar datos del paciente.
- IX. Desplegar farmacias cercanas en donde adquirir los medicamentos.

Requerimiento/Objetivo	Privacidad
Descripción	La información que se ingrese solo la podrá visualizar el doctor que tenga los permisos necesarios otorgados por su paciente.
Razón fundamental	Confidencialidad y seguridad para que el usuario tenga la tranquilidad de usar el

Equipo “Panda-LC”

	sistema.
Implicaciones	Términos y Condiciones tediosas tanto para el paciente como el doctor que se necesitarán actualizar periódicamente para que se encuentren al tanto.

Requerimiento/Objetivo	Seguridad
Descripción	Encriptación de la información almacenada en los servidores.
Razón fundamental	Protección de datos, y que no se permita un mal-uso de los mismos. Así mismo, un acceso a ellos por medio de verificación y validación de información.
Implicaciones	El response time puede ser lento.

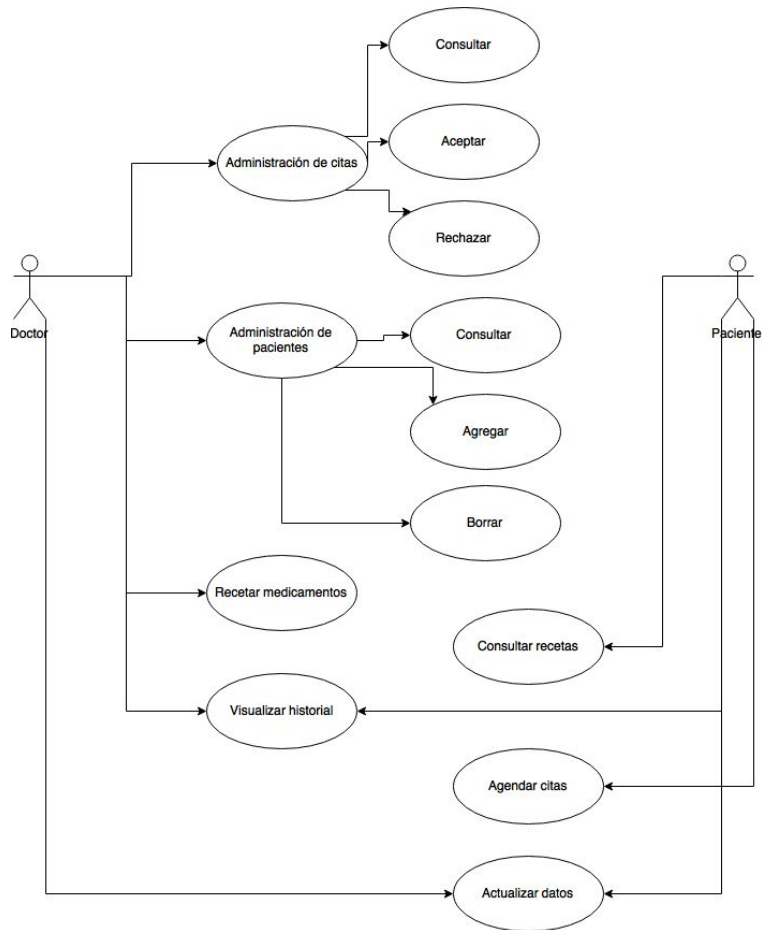
Mi Dientito debe seguir las siguientes restricciones:

Restricción	Seguridad
Descripción	Restricciones de acceso ante cualquier circunstancia como historial, recetas, información personal.
Razón fundamental	Protección de datos.

Equipo “Panda-LC”

Implicaciones	El sistema de seguridad puede ser vulnerable.
---------------	---

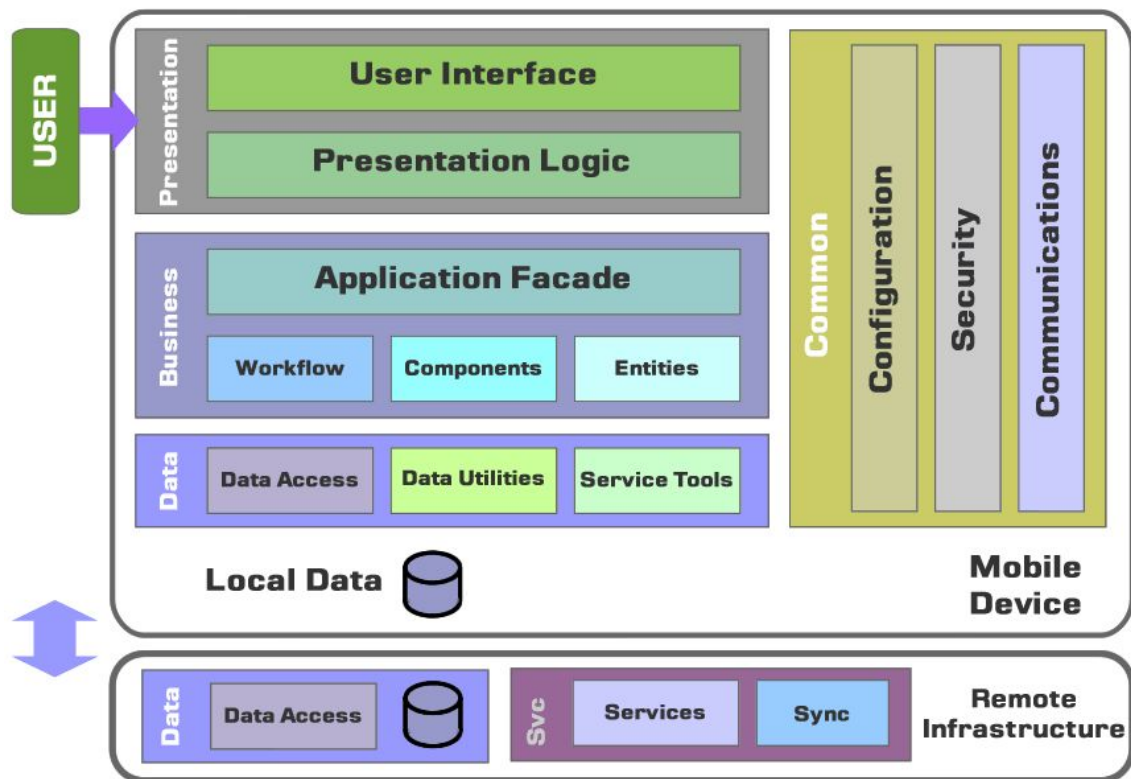
4. Use-Case View



5. Logical View

Overview

Equipo “Panda-LC”

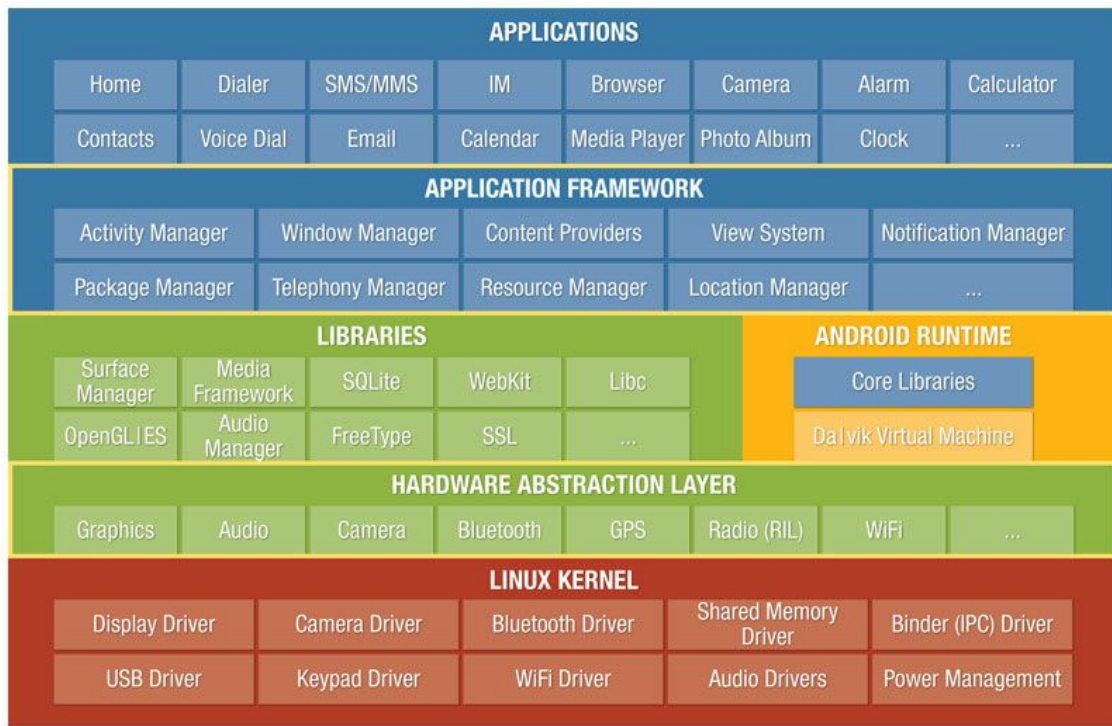


El modelo en capas de “Mi diente” está basado en la estrategia de responsabilidad por capa, en la que a cada capa se le asocia una responsabilidad en particular.

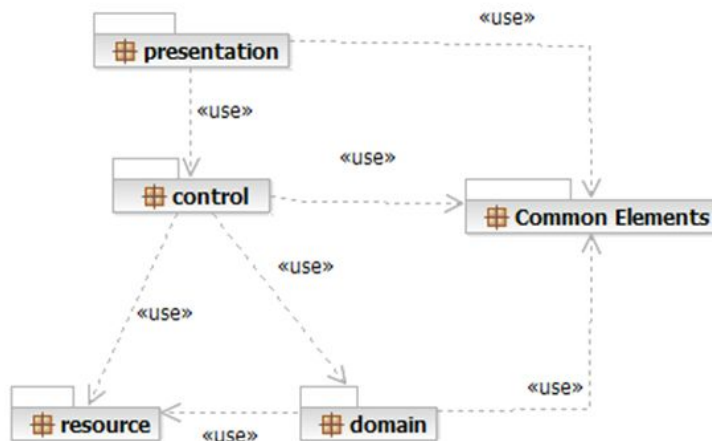
Esta estrategia ha sido elegida porque separa diferentes responsabilidades del sistema una de otra, para así mejorar tanto el mantenimiento como el desarrollo.

La aplicación será desarrollada para Android y a continuación se ilustra la arquitectura de esta plataforma.

Equipo “Panda-LC”



Architecturally Significant Design Packages



La capa de presentación se encarga de renderear las vistas y manejar la lógica de presentación.

Equipo “Panda-LC”

La capa de control maneja el acceso a la capa de dominio.

La capa de recursos es responsable del acceso a la información o los datos que son requeridos por la aplicación.

Los elementos comunes manejan los objetos que son usados por todas las capas.

A continuación se enlistan algunas clases que son importantes en el desarrollo específico de nuestra aplicación:

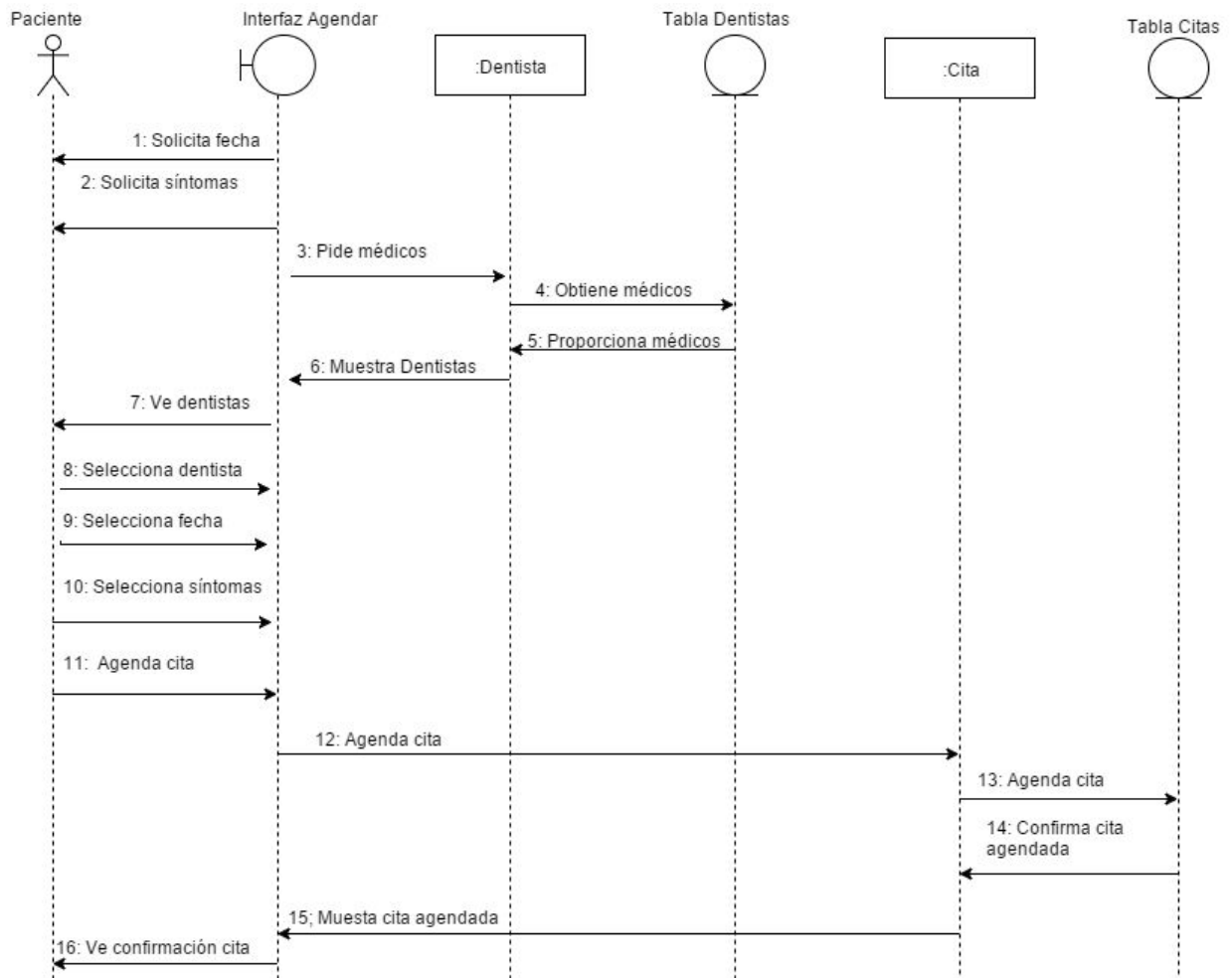
Aplicación paciente	
Login	Para el caso del paciente solo será necesario registrarse una vez y hacer login solo la primera vez que ingrese a la app. Después sus datos serán almacenados localmente y se usarán para los inicios de sesión subsecuentes.
Medicina	Ayudará a tener objetos de este tipo, para que el paciente pueda ver aquellos que le fueron recetados.
Farmacia	El paciente podrá ver un listado de farmacias cercanas a su ubicación en donde podrá adquirir las

Equipo “Panda-LC”

	medicinas que le fueron recetadas.
Recordatorios	El paciente podrá ver sus próximas citas.
Aplicación dentista	
Login	Permite al dentista ingresar al sistema, se requiere de usuario y contraseña.
AgregarPaciente	El dentista puede agregar nuevos pacientes.
HacerReceta	Permite al dentista generar una receta al paciente.
PerfilPaciente	Muestra los datos del paciente que ha sido seleccionado.
Medicamentos	Enlista los medicamentos que puede recetar al paciente en cuestión, para que sean agregados a su receta.
Pacientes	Enlista los pacientes que tiene registrado el dentista.

Equipo “Panda-LC”

El siguiente diagrama de secuencia muestra el proceso que se seguirá para que un paciente pueda agendar una cita:



Use-Case Realizations

Casos de uso:

- Administración de citas: Un dentista será capaz de administrar las citas que tenga con sus pacientes; podrá consultarlas, confirmarlas o, de ser necesario, cancelarlas.
- Administración de pacientes: El responsable del manejo de pacientes será

Equipo “Panda-LC”

el Dentista. Siendo él quien podrá ver su lista de pacientes, agregar o eliminar.

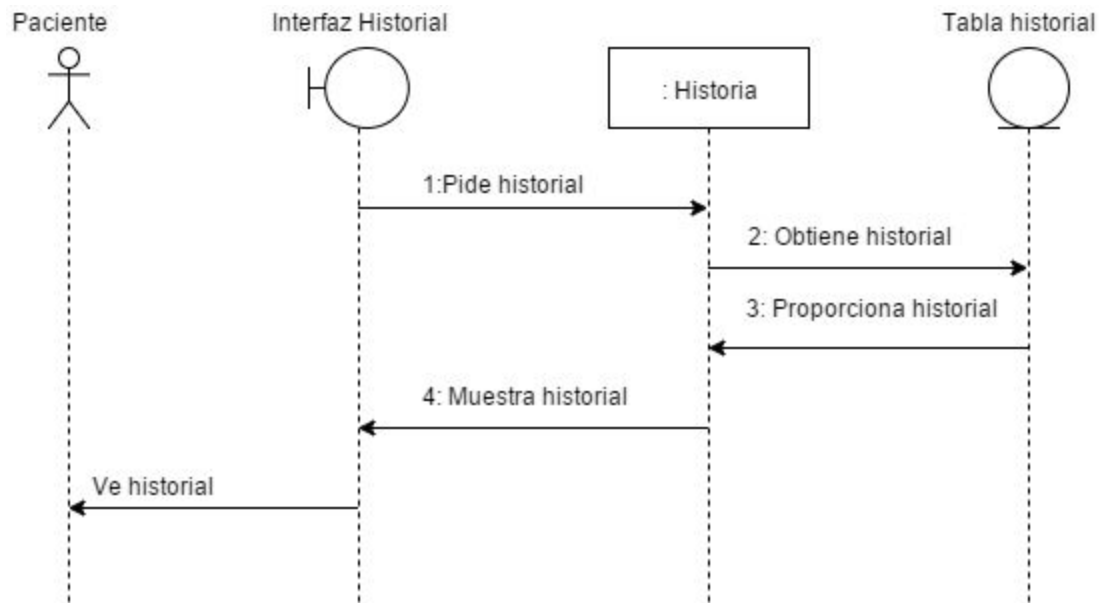
- Recetar medicamentos: El dentista, de ser necesario, podrá recetar medicamentos al paciente o dar indicaciones a seguir para un tratamiento.
- Consultar recetas: El paciente podrá consultar los medicamentos recetados y e indicaciones de su médico.
- Visualizar historial: Tanto el dentista, como el paciente, podrán visualizar el historial de recetas.
- Agendar citas: El paciente podrá agendar una cita con su respectivo dentista, quien, posteriormente aprobará o rechazará la solicitud.

6. Process View

A continuación se muestran diagramas de secuencia con los 4 procesos que fueron considerados como los más relevantes para ilustrar su funcionamiento:

Equipo “Panda-LC”

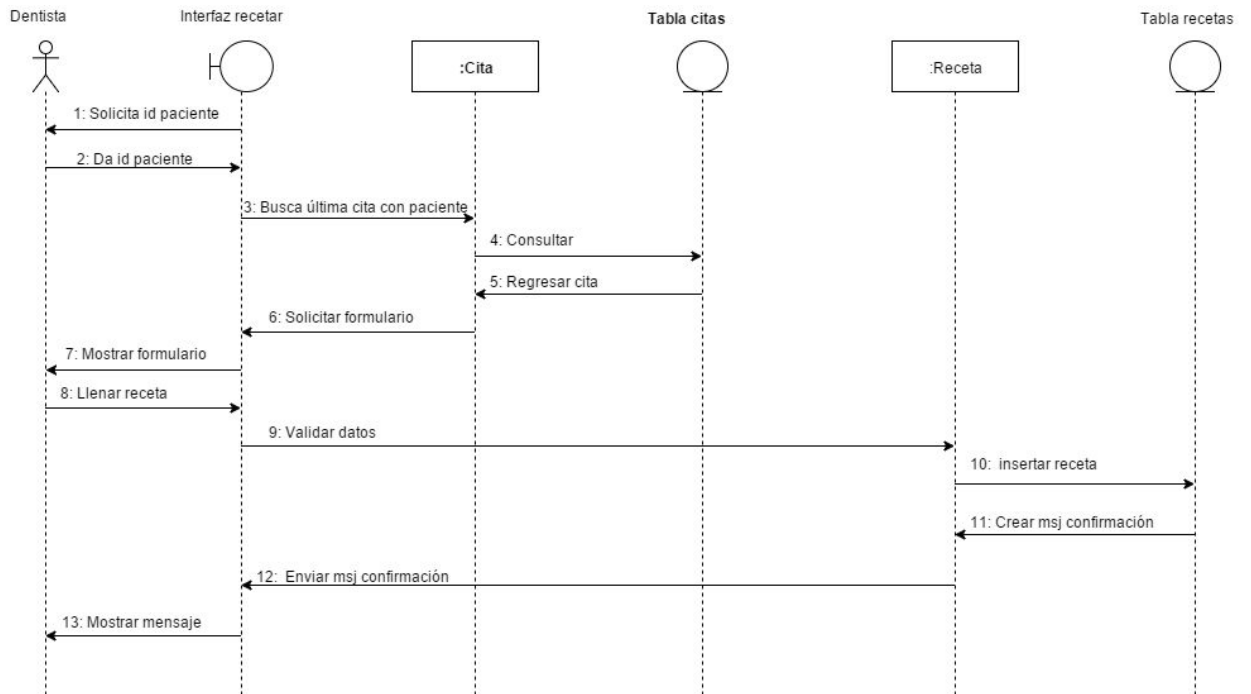
Diagrama de secuencia “Ver Historial”



El diagrama anterior corresponde al proceso que sigue un paciente para ver su historial. El paciente ingresará a la Actividad “Historial” y se consultará en la Base de datos la tabla Historial Médico para consultar todas las recetas que el dentista le ha recetado, basándose en el id. El Paciente podrá ver las recetas en una lista por fechas.

Equipo “Panda-LC”

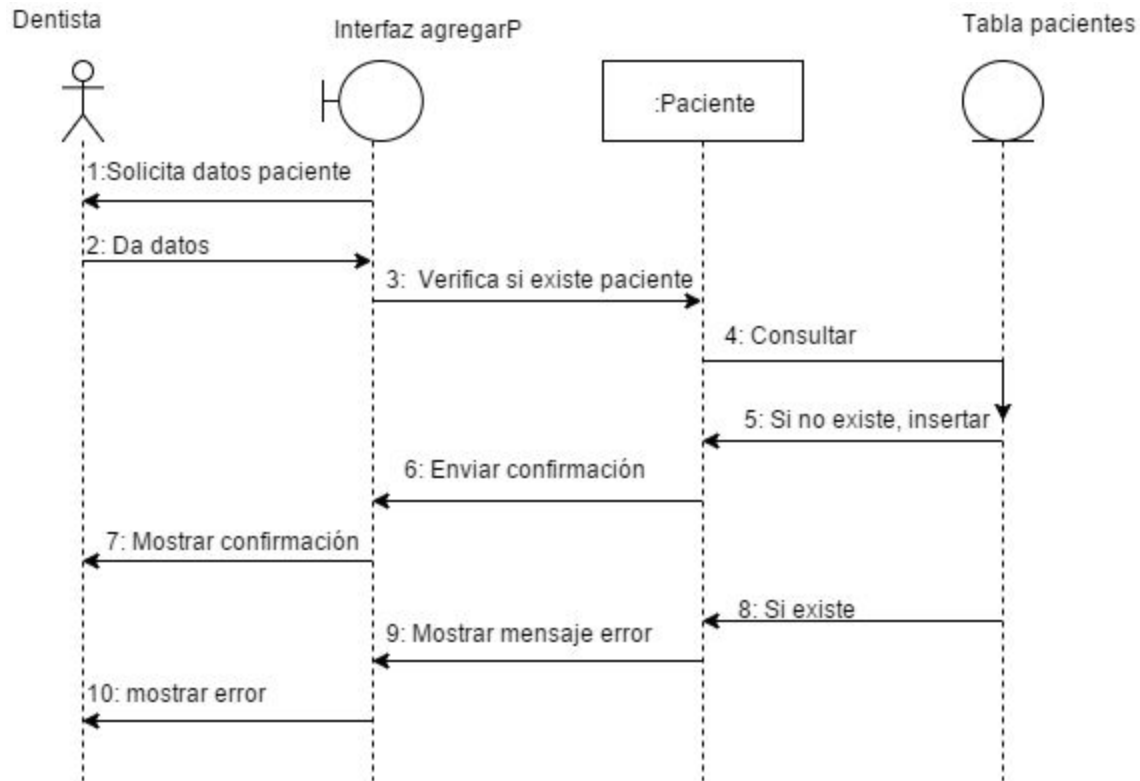
Diagrama de secuencia “Recetar”



Este diagrama muestra el proceso que seguirá un dentista para poder recetar algo a un paciente. La actividad solicitará el id del paciente y buscará en la BD, en la tabla citas, la última cita con el paciente, para que el médico pueda ver datos de ésta y posteriormente pueda recetar un medicamento o dar una indicación.

Equipo “Panda-LC”

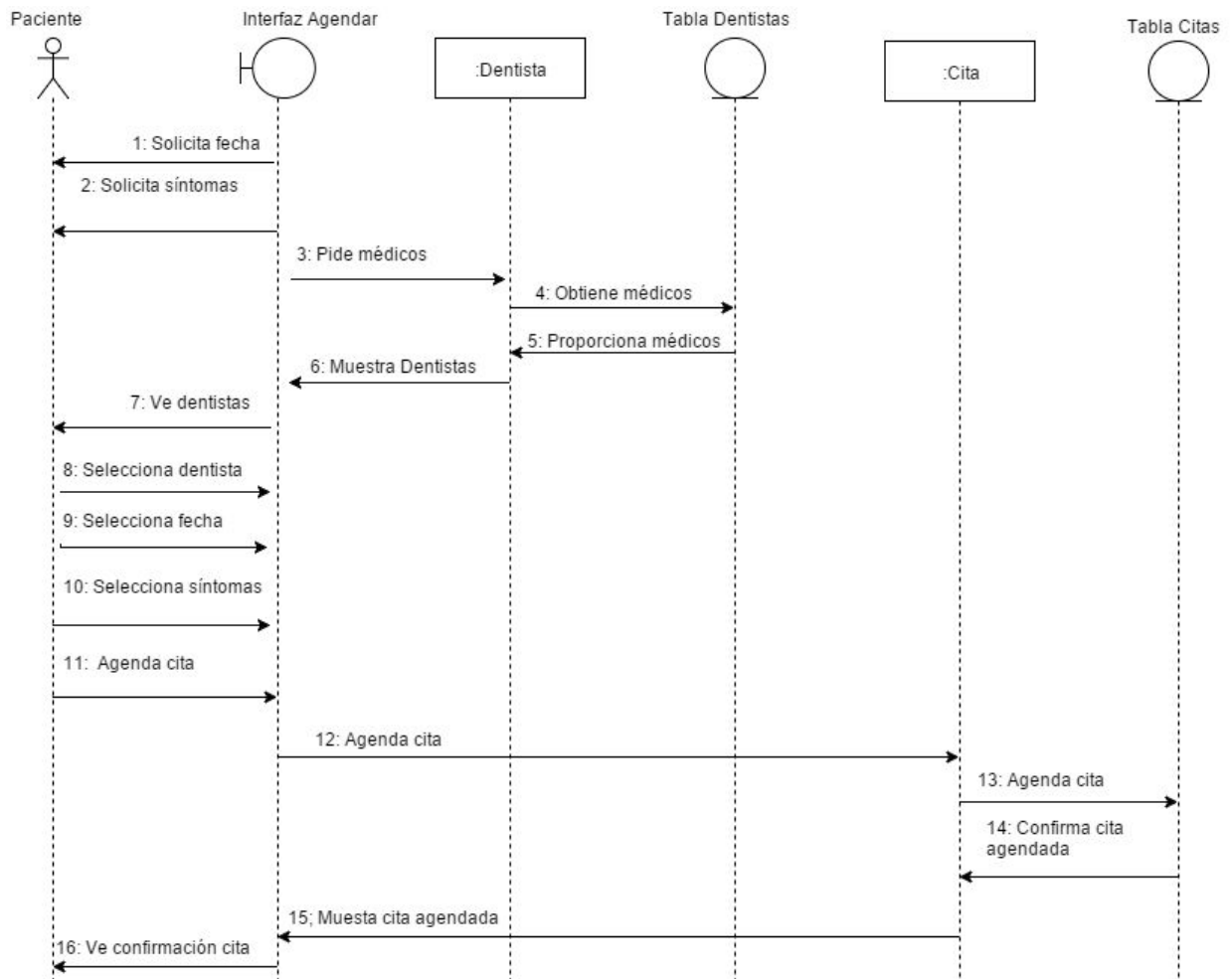
Diagrama de secuencia “Agregar paciente”



En el diagrama anterior se muestra el proceso de dar de alta a un paciente. Para ello, un dentista debe dar los datos de un paciente. Se consulta en la base de datos, en la tabla Pacientes, si el paciente ya existe. Si existe, se muestra un mensaje de error, sino, se inserta.

Equipo “Panda-LC”

Diagrama de secuencia “Agendar cita”



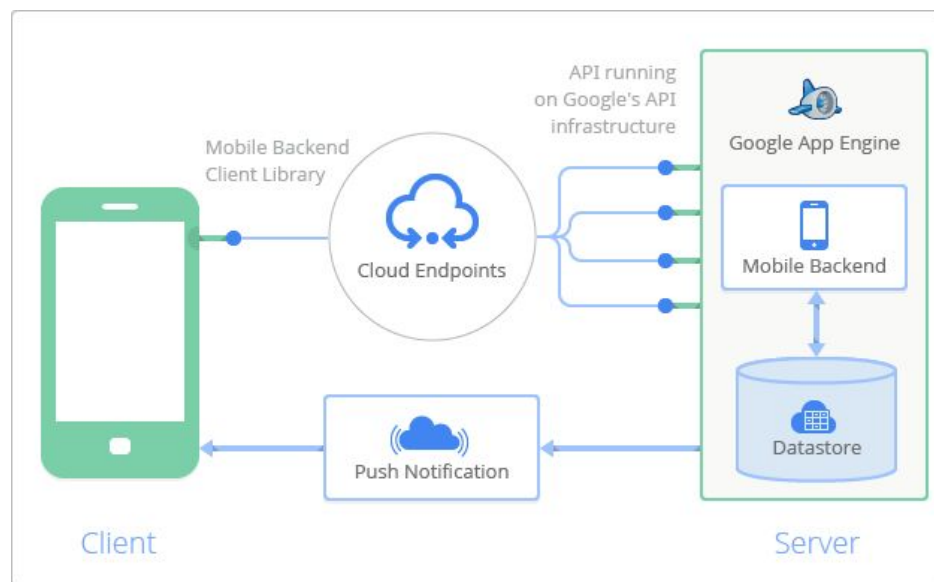
El diagrama anterior muestra el proceso de agendar una cita. Se solicitará al paciente los datos pertinentes para agendar una cita (fecha, síntomas, el id del médico), posteriormente el paciente podrá agendar una cita. Si los datos son validados correctamente, se enviará un mensaje de confirmación, notificando que la cita ha sido agendada.

Equipo “Panda-LC”

7. Deployment View

El manejo de nuestros datos en la aplicación será responsabilidad de la plataforma Parse, que es un backend que provee herramientas dentro de un servidor web, para que así podemos implementar algunas funcionalidades en nuestra aplicación.

Para poder acceder a los datos guardados en el servidor desde nuestra aplicación, desarrollaremos servicios en nuestra aplicación que permitirán que mediante la conexión que se establece con Parse el envío y recibimiento de datos sea sencillo.



Las herramientas SDK de android nos permiten compilar y empaquetar el código junto con cualquier información requerida, y los recursos en archivos de Android con el sufijo .apk. Los archivos .apk representan una aplicación de Android que será desplegada en los dispositivos móviles que tengan Android.

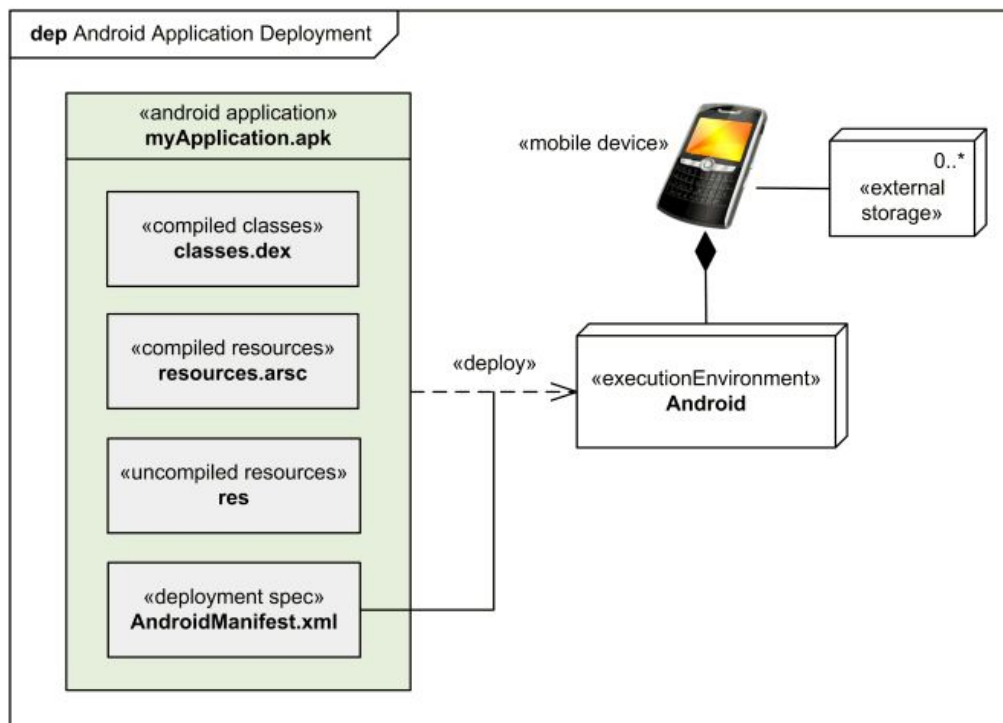
Las aplicaciones Android están constituidas de uno o más componentes de

Equipo “Panda-LC”

la aplicación (actividades, servicios, proveedores de contenido, etc.). Cada componente realiza una función diferente en el comportamiento general de la aplicación, y cada uno se puede activar de forma individual.

El AndroidManifest.xml describe los requisitos de la aplicación (la versión mínima requerida de Android y las configuraciones de hardware compatibles) y en él también se declaran todos los componentes de la aplicación.

La aplicación puede estar instalada en el almacenamiento externo (por ejemplo, en la tarjeta SD). Esta es una característica opcional que puede ser especificada como atributo en el manifiesto. Por defecto, la aplicación se instala en el almacenamiento interno del dispositivo móvil.



8. Implementation View

Overview

La vista de la implementación representa la composición física de la aplicación en términos de implementación de subsistemas y de elementos como directorios y archivos, código fuente, datos y archivos ejecutables.

Por lo general, las capas de la vista de la implementación encajan en la vista lógica.

No es necesario documentar la vista de la implementación a gran detalle en este documento.

Layers

- Capa de Presentación: ésta contiene todos los componentes que son indispensable para permitir la interacción con un usuario final; es decir, la interfaz del usuario.

Capa de presentación:

1. Identificar el tipo de cliente:

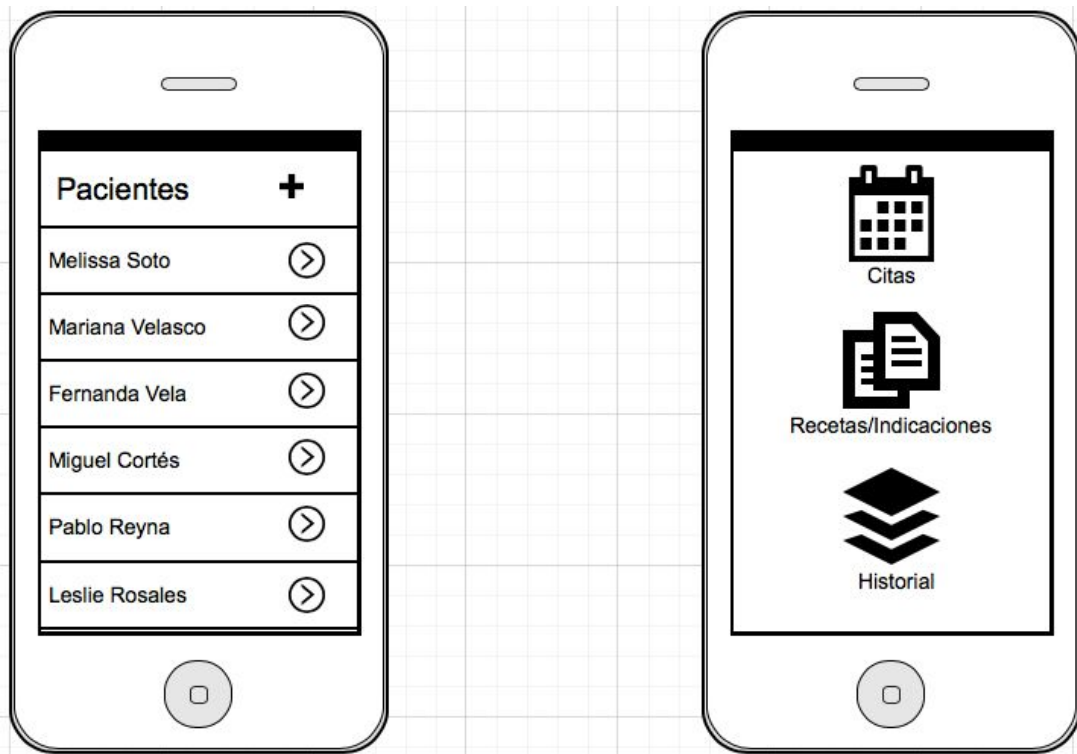
El cliente de MiDientito serán todos los dentistas que necesiten ayuda para organizarse mejor con respecto a sus citas o que busquen una forma más rápida y eficiente de tomar notas clínicas, así como una mejor comunicación con sus pacientes. Por otro lado, los clientes también serán aquellos pacientes que deseen poder tener acceso a su historial médico y que busquen agendar citas de manera rápida y sin contratiempos, proporcionando sólo el motivo de su cita y la fecha en que mejor le convenga.

Equipo “Panda-LC”

2. Diseñar navegación e interfaz de usuario modular.



Equipo “Panda-LC”



Equipo “Panda-LC”

The image displays two smartphone screens side-by-side, representing a dental management application. Both screens have a 'Back' button in the top left corner.

Left Screen: Nuevo Paciente

This screen is used for adding a new patient. It contains the following input fields:

- Nombre
- Edad
- Sexo
- Sangre
- Contacto Emergencia
- Dirección
- Teléfono
- Observaciones

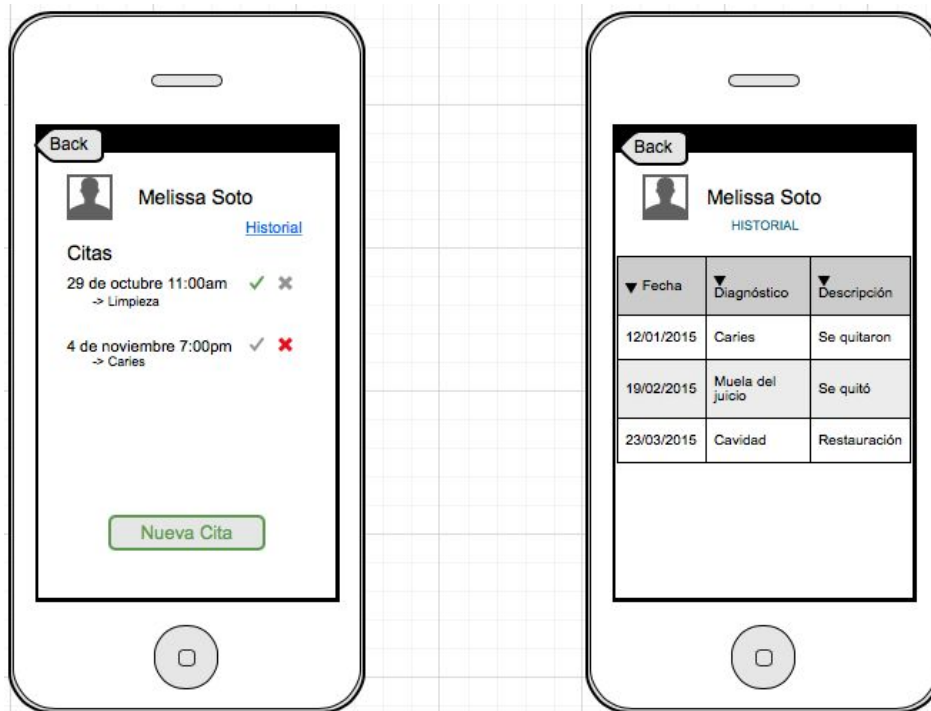
An 'Agregar' button is located at the bottom right of the form.

Right Screen: Citas

This screen displays a list of appointments. The appointments are:

- Limpieza - 29/10/2016
- Caries - 31/10/2016
- Resina - 21/11/2016

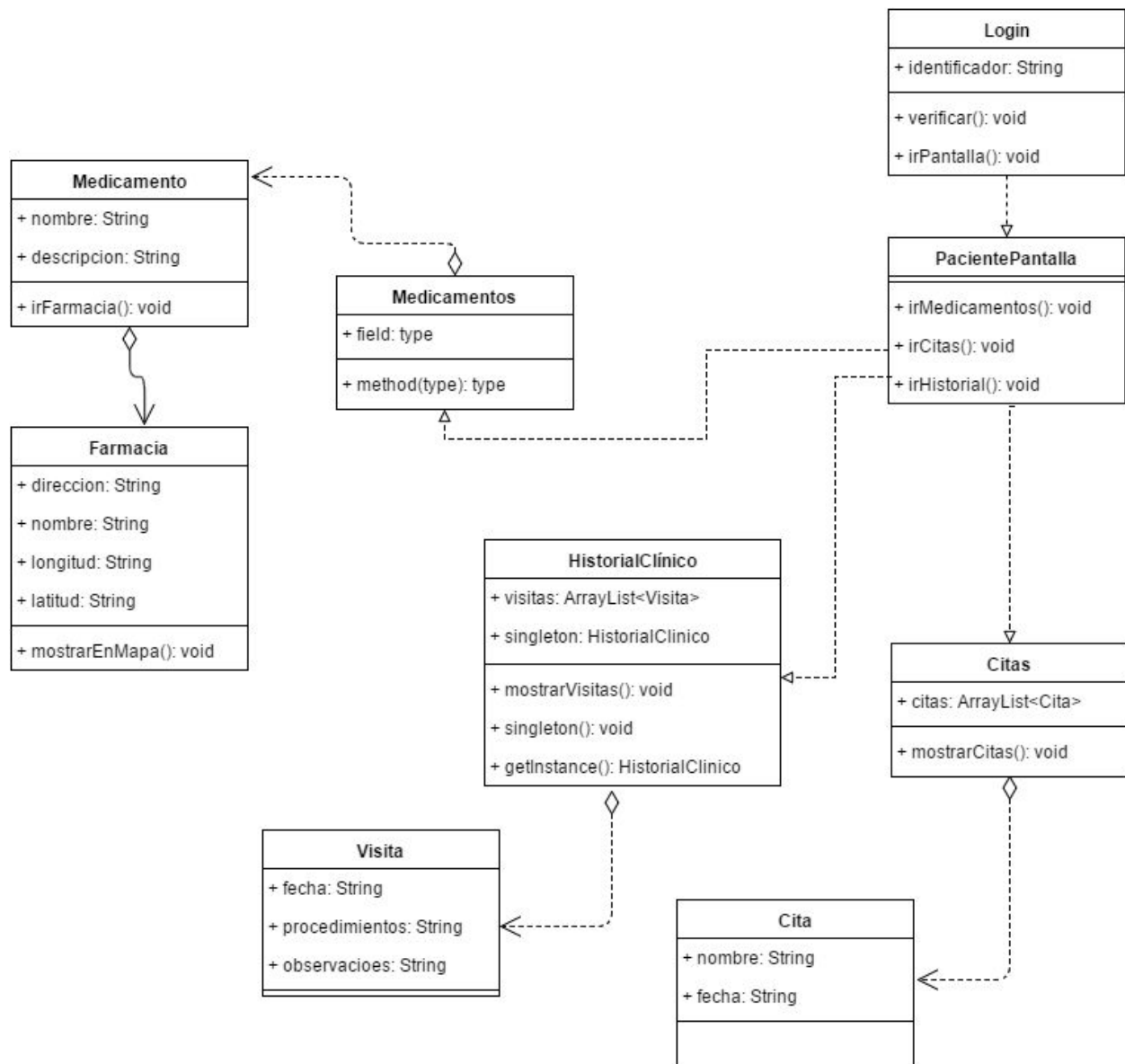
Equipo “Panda-LC”



3. Definir representación de objetos de la vista equivalentes al modelo de negocio.

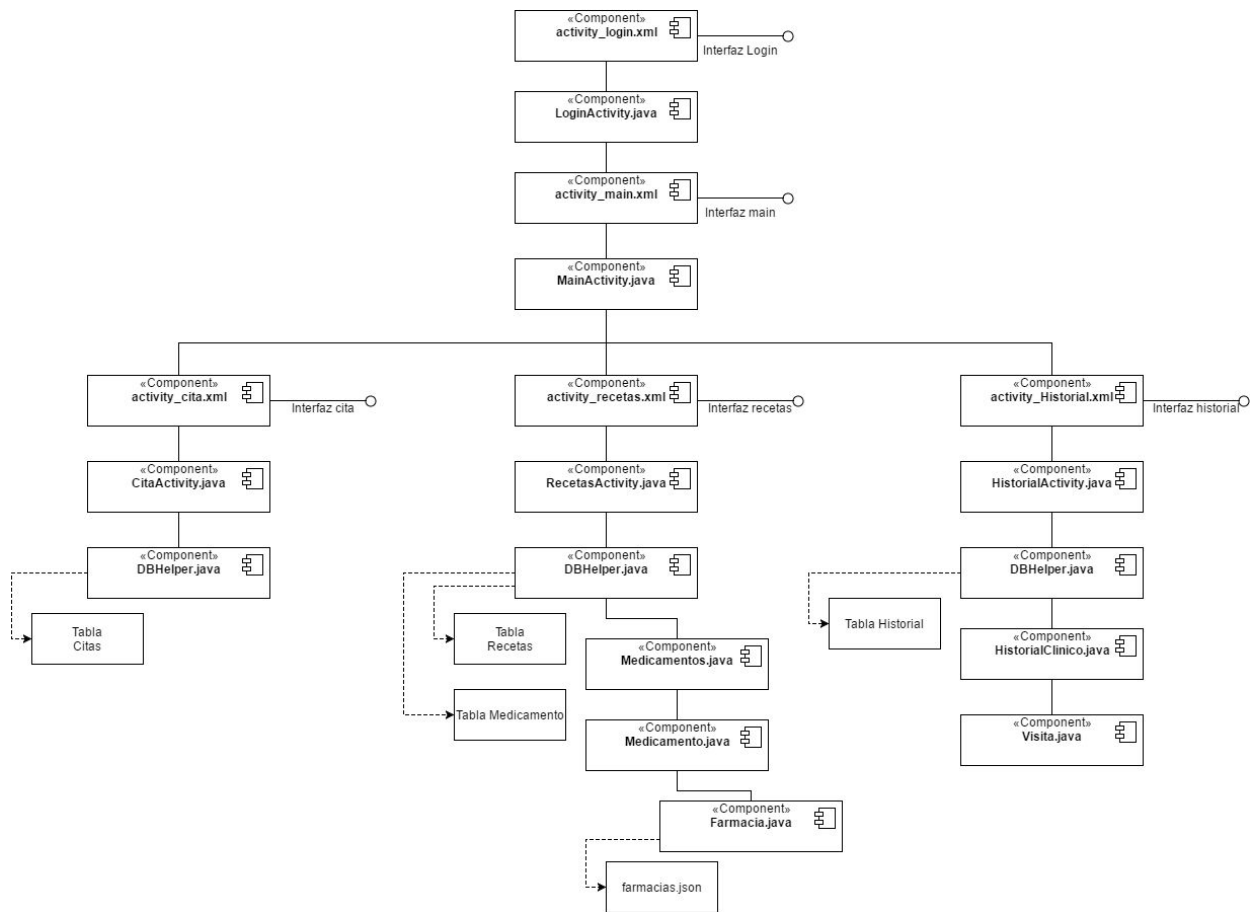
El siguiente diagrama corresponde al diagrama de clases de la aplicación para pacientes. La clase Login permitirá al usuario acceder al sistema, y una vez identificado podrá ingresar a la pantalla principal que lo dirigirá a sus Citas, Medicamentos e HistorialClínico.

Equipo “Panda-LC”



El diagrama que se muestra a continuación corresponde al diagrama de Componentes de la aplicación para usuarios. Se observa que el primer componente es la actividad Login, donde el usuario iniciará sesión y accederá a Main Activity. Desde esta actividad podrá ir a uno de los tres componentes principales del sistema: Actividad Cita, Actividad Recetas o Actividad Historial.

Equipo “Panda-LC”



4. Responder cada una de las consideraciones del Marco de Arquitectura.

- Caching:** Los datos que van a poder permanecer en caché del cliente serán las citas médicas que tenga, ya que son consultadas frecuentemente. Así mismo, otro dato será el ID del usuario para iniciar sesión, ya que de igual forma es un dato al que se accede mayormente para hacer todas las consultas del sistema. En este último caso las transacciones serán por medio de cookies, ya que se podrá llevar el control de usuarios para no estar ingresando sus datos para cada vista.

Equipo “Panda-LC”

- b. Validaciones: La validación de datos se hará desde el servidor, para comprobar que cualquier dato proporcionado por el usuario cumpla con los requisitos para que sea aceptado. Se validarán todos los campos, para evitar que un usuario pueda insertar un dato vacío o erróneo a la base de datos y corromperla. Esta validación se realizará una vez que un formulario ha sido enviado y mostrará un mensaje de error al usuario para que pueda corregir los datos proporcionados y reenvíe el formulario.
- c. Restricción y control de acceso. Existen dos aplicaciones que están limitadas para el doctor y el paciente. Los doctores registrados en el sistema pasarán por un proceso de aceptación en el que se verificará, mediante su cédula profesional, que se trata de dentistas titulados, y únicamente ellos podrán iniciar sesión a la aplicación de Dentistas. Estos dentistas son los únicos que podrán agregar pacientes al Servidor y se les asignará un identificador con el cual podrán iniciar sesión en la aplicación “Mi Dientito”.
- d. Comunicación con la capa de negocio: En la parte chunky lo involucramos en login, ya que el método logon() va a traer username y password que forman parte de un solo método, entonces se disminuye las llamadas al servidor. Para evitar tener transacciones duplicadas, se comprobará que, antes de poder insertar una tupla en la base de datos, que no haya otra tupla idéntica a ésta. Así se podrá garantizar, por ejemplo, que un usuario no pueda agendar la misma cita múltiples veces. Además, se limpiarán todos los campos una vez que una inserción sea completada. Por otra parte, si un dentista desea aceptar/rechazar una cita, una vez que ejecute la acción, será imposible volver a la actividad donde se de la opción de aceptar o

Equipo “Panda-LC”

rechazar una cita, el dentista solo podrá ver el estatus de la cita.

- e. Navegación. En todas las pantallas de la aplicación se podrá regresar a la página de inicio, en el caso de “Mi dientito” la página inicial constará de iconos que le permitirán al usuario navegar por las diferentes secciones de la app (Navegación lineal). En la aplicación del dentista se mostrará una lista con los pacientes y al seleccionar uno entrará a un menú con las opciones propias de cada paciente como modificar sus datos, ver su historial, agendarle una cita, etc (Navegación jerárquica). Se utilizará letra visible y colores sólidos.
- f. Manejo de sesiones. En la aplicación del dentista, éste deberá iniciar sesión cada vez que abra la aplicación, ingresando su identificador y contraseña; esto debido a los datos personales de los pacientes que estarán disponibles en la aplicación. Mientras que en la aplicación del paciente, solo es necesario iniciar sesión la primera vez que tenga acceso a ella. Además en el servidor habrá registro de la fecha en la que fue creado cada usuario.
- g. Desempeño: No se espera que los resultados de la capa de negocios sean realmente altos, dado que MiDientito es una aplicación para un solo dentista y sus pacientes. El número aproximado de pacientes al día es de 30 en un inicio, por lo que las consultas realizadas no traerán grandes cantidades de datos. Se podría limitar el tamaño del historial clínico a los últimos cinco años, para no tener resultados muy grandes.

5. Selección de tecnología:

Se hará uso del sistema operativo Android ya que se puede instalar en la mayoría de los dispositivos del mercado actual, por lo que llegará a un mayor público. Así mismo, se cuenta con la comunidad más grande a nivel

Equipo “Panda-LC”

mundial de desarrolladores, es decir, más foros para solucionar dudas. Con respecto a las capacidades de almacenamiento y servidores que usaremos, éstos son los mínimos necesarios para hacer funcionar el sistema, quitando muy poco espacio del celular del usuario y robando muy pocos datos cuando se ingresa al servicio. Estará hosteado en Parse ya que ofrece almacenamiento de una gran cantidad de datos a un precio considerable, así como un acceso instantáneo a la información.

6. Aplicar patrones de diseño.

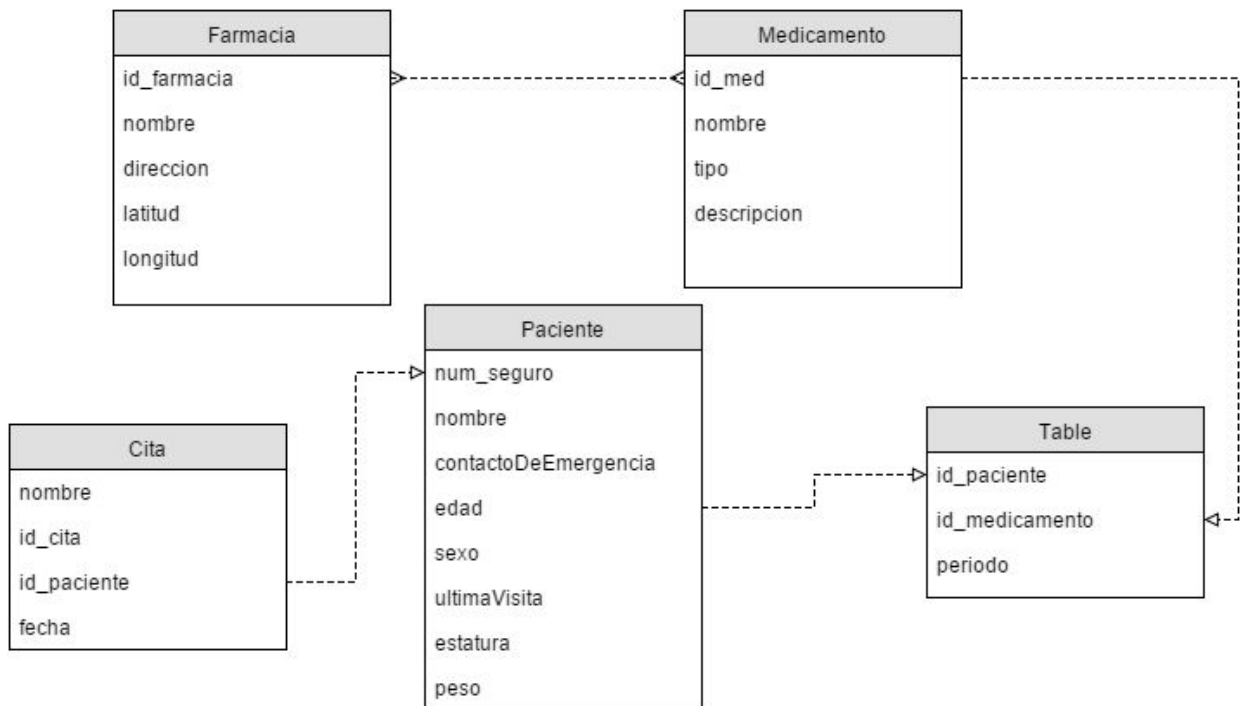
Un patrón de diseño significa añadir capas de abstracción; es decir, cuando se abstrae algo se aíslan detalles concretos. Su objetivo es encapsular los cambios. Los siguientes son algunos de los que se implementan en el proyecto:

1. Singleton: este patrón de diseño se usa como un solo administrador de pantalla; es decir, permite centralizar la gestión y provee un acceso global al mismo. El caso sería con Historial, ya que solamente existe una clase que se instancia a sí misma para prevenir que se hagan más.
 2. Factory Pattern: este patrón de diseño crea objetos sin exponer la lógica al cliente. El caso sería con crearPaciente, ya que se registra un nuevo producto (paciente) a la fábrica sin cambiar nada de ella.
 3. Prototype: para la creación de recetas, ya que son similares y solo cambian en la descripción. Y de igual forma, la creación de citas.
- Capa de Control: ésta contiene todos los componentes utilizados para acceder a la capa de dominio o directamente a la capa de recursos cuando sea apropiado.

Equipo “Panda-LC”

- Capa de Recursos: ésta contiene los componentes que se necesitan para permitir una comunicación entre el segmento de negocio y servicios como ERP.
- Capa de Dominio: ésta contiene todos los componentes que se relacionan con la lógica del negocio. Recopila todos los subsistemas que satisfagan las necesidades de un dominio de negocio en particular. Incluso, tiene el modelo de objetos de negocio.
- Capa de Elementos en Común: ésta contiene todos los componentes que se reutilizan dentro de varias capas.

9. Data View (optional)



10. Size and Performance

[A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.]

Equipo “Panda-LC”

Volumen:

1. Número estimado de citas: 100 al día
2. Usuarios registrados al mes(dentistas): 5
3. Usuarios registrados al mes(pacientes): 50
4. El dispositivo del cliente requiere de al menos 60 MB

Desempeño:

Tiempo de procesamiento por transacción: se requiere que sean menos de 10 segundos

La arquitectura seleccionada soporta los requerimientos de tamaño y tiempo, además es importante recalcar que la porción del cliente está implementada en dispositivos móviles. Los componentes han sido diseñados para asegurar que la mínima cantidad de requerimientos de memoria y disco sean necesarios en el dispositivo de la parte cliente.

11. Quality

El sistema obedecerá diferentes atributos de calidad, entre los que se encuentran:

Escalabilidad:

Descripción: Cómo reacciona el sistema cuando la demanda incrementa

Solución: El servidor soporta diferentes técnicas para el manejo de la carga de trabajo.

Disponibilidad, Confiabilidad

Descripción: Mecanismo contra fallas transparente, tiempo medio

Equipo “Panda-LC”

entre fallos

Solución: El servidor soporta balanceo de cargas a través de clusters

Portabilidad

Descripción: Habilidad para ser reusado en otro ambiente.

Solución: Se puede implementar y desplegar en otras aplicaciones que usen el mismo servidor.

Seguridad:

Descripción: Mecanismos de autenticación y autorización

Solución: El servidor ofrece mecanismos de seguridad nativos, que serán usados.