

Ministerio de Producción

Secretaría de Industria y Servicios

Subsecretaría de Servicios Tecnológicos y Productivos

y

Ministerio de Educación y Deportes



Analistas del Conocimiento

Dimensión Programador

Guía de Ejercicios Prácticos para el Módulo

Desarrollo de Software

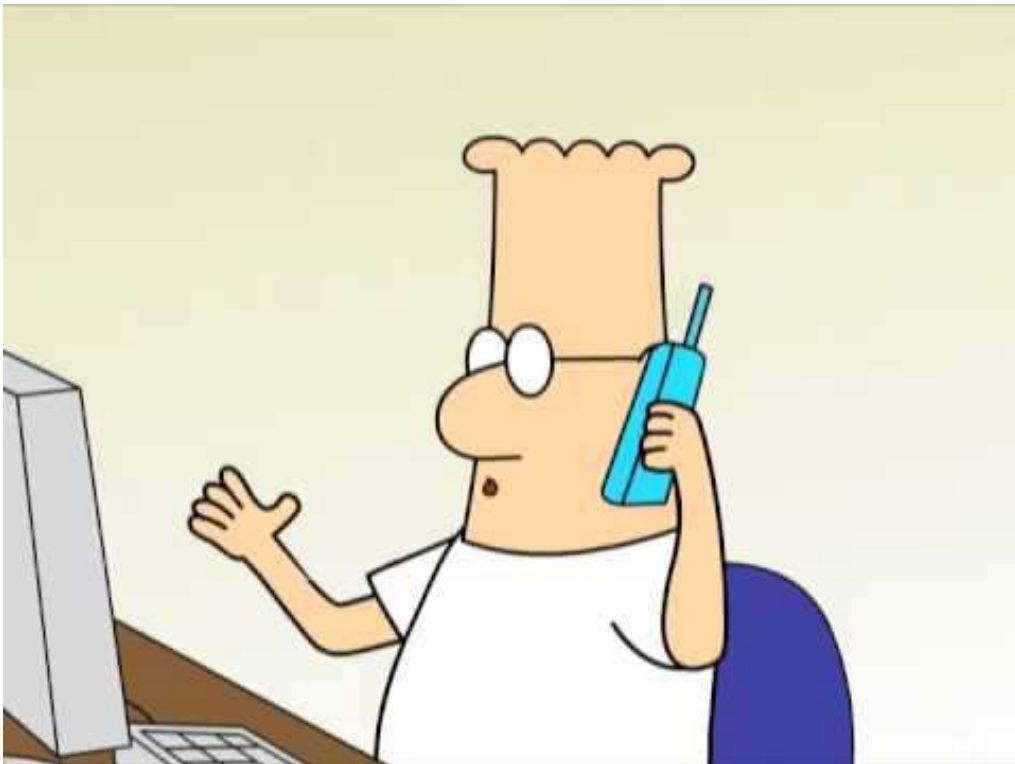


Tabla de Contenido

INTRODUCCIÓN	5
ENUNCIADOS DE LOS EJERCICIOS A DESARROLLAR EN LA GUÍA	6
TEMA: USER STORIES.....	6
CONSIGNA PARA LOS EJERCICIOS DE ESTE TEMA:	6
EJERCICIO 1: CASO PRÁCTICO – FESTIVAL DE FOLKLORE	6
EJERCICIO 2: CASO PRÁCTICO – TAXI MOBILE: SISTEMA WEB MOBILE PARA SEGUIMIENTO DE TAXIS	7
EJERCICIO 3: CASO PRÁCTICO – SISTEMA WEB MOBILE PARA REGISTRO DE GASTOS.....	8
EJERCICIO 4: CASO PRÁCTICO – SISTEMA GPS	9
TEMA: REVISIONES TÉCNICAS AL SOFTWARE	11
CONSIGNA PARA LOS EJERCICIOS DE ESTE TEMA:	11
EJERCICIO 1: CASO PRÁCTICO – SISTEMA GPS	12
EJERCICIO 2: CASO PRÁCTICO – SISTEMA DE AFLUENCIA	13
EJERCICIO 4: CASO PRÁCTICO – DESARROLLOS A MEDIDA	16
TEMA: ADMINISTRACIÓN DE CONFIGURACIÓN DE SOFTWARE	17
CONSIGNA PARA LOS EJERCICIOS DE ESTE TEMA:	17
EJERCICIO 1: CASO PRÁCTICO – GPS	17
EJERCICIO 2: CASO PRÁCTICO – HERRAMIENTA DE SEGUIMIENTO DE DEFECTOS.....	18
EJERCICIO 3: CASO PRÁCTICO – LIGA REGIONAL DE FÚTBOL	19
EJERCICIO 4: CASO PRÁCTICO – SOFTWARE PARA CALL CENTER	21
SOLUCIONES PROPUESTAS.....	24
TEMA: USER STORIES.....	24
EJERCICIO 1: CASO PRÁCTICO – FESTIVAL DE FOLKLORE	24
SOLUCIÓN PROPUESTA	25
TEMA: USER STORIES.....	31
EJERCICIO 2: CASO PRÁCTICO – TAXI MOBILE: SISTEMA WEB MOBILE PARA SEGUIMIENTO DE TAXIS	31
TEMA: USER STORIES.....	37
EJERCICIO 3: CASO PRÁCTICO – SISTEMA WEB MOBILE PARA EL REGISTRO DE GASTOS.....	37
SOLUCIÓN PROPUESTA	38
TEMA: USER STORIES.....	41
EJERCICIO 4: CASO PRÁCTICO – SISTEMA GPS	41
TEMA: REVISIONES TÉCNICAS AL SOFTWARE	45
EJERCICIO 1: CASO PRÁCTICO – SISTEMA GPS	45
SOLUCIÓN PROPUESTA:	46
EJERCICIO 2: CASO PRÁCTICO – SISTEMA DE AFLUENCIA	47
PROPUESTA DE SOLUCIÓN	50
EJERCICIO 3: CASO PRÁCTICO – EMPRESA DE TELEVISIÓN POR CABLE.....	51
SOLUCIÓN PROPUESTA	53
EJERCICIO 4: CASO PRÁCTICO – DESARROLLOS A MEDIDA	54
SOLUCIÓN PROPUESTA:	55
TEMA: ADMINISTRACIÓN DE CONFIGURACIÓN DE SOFTWARE	56
EJERCICIO 1: CASO PRÁCTICO – GPS	56
CONSIDERACIONES PARA EL PROYECTO	56

PROPUESTA DE SOLUCIÓN.....	57
EJERCICIO 2: CASO PRÁCTICO – HERRAMIENTA DE SEGUIMIENTO DE DEFECTOS.....	57
PROPUESTA DE SOLUCIÓN.....	60
EJERCICIO 3: CASO PRÁCTICO – LIGA REGIONAL DE FÚTBOL	64
PROPUESTA DE SOLUCIÓN.....	65
EJERCICIO 4: CASO PRÁCTICO – SOFTWARE PARA CALL CENTER	67
PROPUESTA DE SOLUCIÓN:.....	70
<u>FUENTES DE INFORMACIÓN</u>	<u>72</u>

Introducción

La guía práctica del Módulo Desarrollo de Software contiene casos prácticos para algunos de los temas que se han desarrollado en el Apunte Teórico del Módulo. Los temas son

- ↳ User Stories y estimación con story points.
- ↳ Revisiones Técnicas de Código
- ↳ Administración de Configuración de Software: identificación de ítems de configuración y estructura de repositorio.

Para cada uno de estos temas se presentan 4 casos de estudio y la consigna asociada a cada tema. La primera parte de la guía contiene un compendio de los casos y luego se presentan los casos con las soluciones propuestas.

Enunciados de los Ejercicios a Desarrollar en la Guía

Tema: User Stories

Consigna para los ejercicios de este tema:

Se pide:

- Asumiendo un rol de Product Owner (PO) realice:
 - La identificación de roles que serán usuarios del sistema.
 - La descripción completa de al menos 5 user stories.

Ejercicio 1: Caso Práctico – Festival de Folklore

Presentación del Caso de Estudio

Anualmente la Dirección de Cultura de la Municipalidad de una localidad de la provincia, organiza un festival de folklore. Este festival tiene una duración de generalmente cinco noches, aunque esto puede variar de año en año. En cada una de las noches actúan distintos grupos folklóricos con reconocimiento regional, provincial y nacional. El festival se prepara con mucha anticipación y se realiza la diagramación para determinar qué grupos actúan en cada noche y el orden en el que los mismos realizarán sus presentaciones, teniendo en cuenta que los horarios de presentación de los grupos no pueden superponerse y que no pueden quedar espacios sin ninguna presentación entre medio de dos grupos. Considerar que no puede incluirse la participación de un grupo más de una vez para un mismo festival, en una misma noche.

En cada noche se define la hora de inicio de la misma, pero no se determina la hora de fin, ya que esta puede variar según si las presentaciones se extienden más de lo previsto.

El Festival se realiza en un único estadio, que está dividido en sectores (A, B, C, etc.), que se identifican con colores diferentes, y cada sector se compone de filas (1, 2, 3, etc.), cada fila, a su vez, está conformada por butacas, las cuales están numeradas.

La venta de entradas se realiza en cinco puntos de venta que se encuentran en funcionamiento simultáneamente: en el estadio donde se realizará el festival, en tres centros comerciales de la ciudad capital y en un centro comercial de la localidad donde se realiza el festival. No se debe permitir que se venda una misma entrada (una misma butaca de un festival en una misma fecha) en dos puntos de venta diferentes.

Existen distintos tipos de entradas para el público (mayores, menores, jubilados, etc.). El precio de las entradas depende del tipo de entrada y del sector donde se encuentre la butaca, además puede variar de una noche a otra, dependiendo de los grupos musicales que actúan. Por ejemplo, una entrada para mayores en el sector A, que está cerca del escenario, será más costosa que una para mayores en el sector E que está más alejado del mismo y a su vez puede variar de noche en noche el precio de la entrada en la misma ubicación. Las butacas se venden para una noche en particular así es que una misma butaca puede estar disponible, por ejemplo, para la noche 1 y 3, y ocupada para la noche 2, 4 y 5.

También se habilita la venta anticipada de las entradas a un precio menor, un porcentaje de descuento que la Dirección de Cultura determina, al igual que la fecha de vencimiento de ese beneficio, por ejemplo, venta anticipada con un descuento del 10 % hasta un mes antes que empiece el festival. La forma de venta de entradas es únicamente de contado en efectivo. Si un cliente solicita la anulación de la entrada sólo se le reintegra el 50% del monto abonado. Esto se puede hacer hasta 10 días antes del inicio del festival.

La entrada tiene un código de barras para evitar falsificaciones. Además, hay que tener en cuenta que la misma entrada cumple la función de factura, por lo que debe tener los datos requeridos por la ley de facturación, y debe asegurarse de que el número de factura sea único.

La Dirección de Cultura de la Municipalidad ha solicitado a su Área de Sistemas el desarrollo de un sistema de información que le ayude con la administración de los festivales que organiza, la diagramación de la programación y la venta de entradas y brinde información que ayude a la organización de próximos festivales. La Dirección de Cultura de la Municipalidad tiene licencias para realizar la aplicación con una base de datos Oracle.

Debido a que en las horas pico se suele generar cola en los puntos de venta, es necesario que el sistema genere una entrada en no más de 6 segundos.

Ejercicio 2: Caso Práctico – Taxi Mobile: Sistema web mobile para seguimiento de taxis

Presentación del Caso de Estudio

Objetivo: Desarrollar un producto de software para Smart phones, donde los pasajeros de taxis puedan solicitar el taxi más cercado y saber su ubicación y demora en todo momento.

A **continuación**, se transcribe parte de la entrevista realizada al experto en el dominio:

Product Owner (PO): ¿La idea es que el pasajero pueda llamar a una central sabiendo dónde está el taxi que va a pedir? ¿Cómo pide un taxi el pasajero?

Experto en el Dominio (ED): El pasajero debe ingresar a la aplicación, la cual activará el sistema de posicionamiento. El sistema detecta dónde está el pasajero y le muestra los 5 taxis más próximos a su ubicación, visualizados en un mapa, e informa ubicación, distancia y tiempo estimado. El pasajero selecciona el taxi deseado y esta acción envía una notificación a la central de taxis y al taxista a su celular. De esta forma ambos identifican el pedido de un móvil, y el taxista puede saber a dónde está el pasajero

PO: ¿Y cómo sabe el taxista quién es el pasajero?

ED: Cuando se instala la aplicación en el celular se pedirán los datos mínimos para registrarlo como usuario. Estos datos se pueden obtener de Facebook y el número de celular del mismo aparato donde se instala la aplicación.

PO: ¿Y los taxistas cómo trabajan con esta aplicación?

ED: Como taxista también es necesario instalar una aplicación en su celular. El taxista debe estar registrado en una central que haya contratado el servicio de taxi-mobile. Cuando instale la aplicación se solicitarán sus datos identificatorios como taxista y de su móvil (nro. de taxi y dominio), y la selección de la central a la que pertenece. La aplicación también utiliza el sistema de posicionamiento que tiene el teléfono, para poder informar en qué lugar se encuentra el taxista.

PO: ¿Si estoy entendiendo bien, tanto el pasajero como el taxista deben contar con un Smart Phone con sistema de posicionamiento para que la aplicación funcione?

ED: Así es, sino la aplicación no puede ubicar al taxi ni al pasajero.

PO: ¿Y una vez que el pasajero sube al taxi, cómo se indica que ya está en viaje?

ED: El taxista debe asentar esto en la aplicación con la opción correspondiente. De esta forma la central está al tanto de la situación. Igualmente, cuando el pasajero descienda el taxista debe indicarlo, para que el taxi quede libre en la aplicación, y se muestre a un próximo pasajero.

PO: ¿La aplicación sólo muestra taxis libres?

ED: A los pasajeros sí, en la central se pueden ver con distintos colores los taxis ocupados, los libres y los que están fuera de servicio.

PO: ¿Y la aplicación para la central también es mobile?

ED: No, debería ser una aplicación web.

PO: ¿La idea es que la aplicación del pasajero y del taxista sean con touch screen¹?

ED: y los elementos centrales de la aplicación son el mapa y el taxi. Siempre a través de la selección del taxi se indicarán las acciones que quieren realizarse, como llamarlo, indicar que está ocupado, que está libre, tanto por parte del pasajero como del taxista.

A continuación, se muestran algunos prototipos de las interfaces que se espera vean los usuarios. La generación de los mapas será provista por la plataforma de Google Maps:



Ejercicio 3: Caso Práctico – Sistema web mobile para registro de gastos

Presentación del Caso de Estudio

Objetivo: Desarrollar un sistema para celulares y Tablet para que los usuarios puedan registrar sus gastos mensuales y mantener un control de los mismos.

A continuación, se transcribe parte de la entrevista realizada al experto en el dominio:

Product Owner (PO): ¿Cómo debe visualizarse la planilla de gastos?

Experto en el Dominio (ED): Similar a una tabla en Excel, con columnas donde se pueda indicar el monto, el tipo de gasto, y la fecha en que se realizó el gasto.

¹ Touch Screen: término en inglés que significa pantalla táctil. Es una pantalla que mediante un toque directo sobre su superficie permite el ingreso de datos y órdenes al dispositivo, y a su vez muestra los resultados introducidos previamente; actuando como periférico de entrada y salida de datos, así como emulador de datos interinos erróneos al no tocarse efectivamente. Este contacto también se puede realizar por medio de un lápiz óptico u otras herramientas similares.

PO: ¿La fecha del gasto es la fecha actual? ¿Se toma automáticamente?

ED: Debería mostrarse la fecha actual, pero permitir modificarse.

PO: ¿Cuáles son tipos de gastos permitidos?

ED: Cada persona debería poder registrar sus tipos de gastos, así como indicar si el gasto es propio o de otra persona (por ejemplo, de su esposa).

PO: ¿El nombre o la relación con el usuario debe indicarse?

ED: Debe indicarse el nombre y apellido. También por defecto debe mostrarse el nombre y el apellido del usuario logueado, permitiendo modificarlo, y si alguna vez se registró el nombre y apellido que se comienza a ingresar (no importa si esta vez o una vez anterior), el sistema debería mostrar aquellos nombres y apellidos que comiencen de forma similar, como hace el Excel.

PO: Ah... ¿Es decir que el usuario debe registrarse y cuando va a usar la aplicación debe iniciar sesión? ¿Y la sesión caduca en algún momento?

ED: El usuario debe registrarse para permitirle descargar la aplicación, y la primera vez debe iniciar sesión, pero luego se consideran los datos de la sesión registrados, salvo que el usuario decida desloguearse. La sesión no caduca nunca.

PO: ¿Y la planilla de gastos muestra todos los gastos? ¿Cómo se ordenan?

ED: Por defecto se deben mostrar todos los gastos del mes en curso ordenados desde el gasto más actual, pero el sistema debería permitir ver cualquier período que el usuario quiera, y que pueda filtrar por tipo de gasto, por responsable de gasto, por rango de montos. Además debe poder modificar el criterio de ordenamiento. Y para cada filtro que se aplique arriba se debe mostrar el total de gastos según el filtro aplicado.

Ejercicio 4: Caso Práctico – Sistema GPS

Presentación del Caso de Estudio

Objetivo: Desarrollar un sistema que permite a un conductor (entre otras funcionalidades), buscar un destino, obteniendo distintas alternativas para llegar hasta el punto marcado desde la ubicación actual.

A continuación, se transcribe parte de la entrevista realizada al experto en el dominio:

Product Owner (PO): ¿Cómo se puede buscar un destino deseado?

Experto en el Dominio (ED): La búsqueda puede realizarse en todos los mapas de las distintas ciudades, que serán cargados en el dispositivo, o bien en el mapa de una ciudad determinada.

PO: ¿Qué datos del destino son necesarios?

ED: Si desea buscar un destino por dirección, debería indicar primero el país y la ciudad, y luego ingresar el nombre de la calle y número.

PO: ¿Qué pasa si desconozco alguno de éstos datos, ¿Puedo buscar el destino por otros parámetros?

ED: Si, también debe existir la posibilidad de buscar un destino mediante sus coordenadas, o indicando un cruce de calles. Las coordenadas se representan con tres números que indican longitud y tres números que

indican latitud. Cada número representa los grados, minutos y segundos respectivamente. Además se debe indicar la orientación para la latitud (norte, sur) y para la longitud (este, oeste), de cada coordenada.

PO: ¿Me darías un ejemplo?

ED: Claro, por ejemplo 24° 45' 45'' Longitud Este – 45° 34' 23'' Latitud Sur.

PO: Gracias, está claro. ¿Mencionaste algo del cruce de calles?

ED: Sí, la búsqueda de un destino podría ser realizada por cruce de calles, primero debería ingresarse el nombre del país y de la ciudad de destino, y luego el nombre de las dos calles.

PO: Entendidos los parámetros de búsqueda, ¿Cómo se espera que se indique el camino?

ED: Se debe poder visualizar en el mapa el camino propuesto para dirigirse desde el punto actual (origen) hasta el destino señalado.

PO: Respecto del camino que debe visualizarse, ¿Alguna condición para las distintas alternativas?

ED: Ciertamente, antes de la visualización en el mapa, debería poder seleccionarse la ruta deseada: el camino más rápido, el camino más corto, el camino por caminos alternativos, el camino evitando peajes o el camino evitando controles. Además, una vez realizada la búsqueda, se debe permitir al conductor marcarla como favorita, ingresando si desea un nombre descriptivo, para que en caso de volver a necesitarla, evitar ingresar todos los datos nuevamente.

PO: ¿Para cualquier tipo de búsqueda?

ED: Así es, una vez encontrado el destino, se debe poder guardar el mismo en Favoritos, ya sea una dirección, un cruce de calles o coordenadas. Cuando el conductor desee dirigirse a un destino guardado con anterioridad, sólo debe consultar la opción Favoritos y buscar el destino deseado. Los destinos en Favoritos deberían visualizarse por orden alfabético según su nombre, y también deberían poder filtrarse deletreando el mismo. Para éste último se desea que a medida que el conductor ingrese el nombre, el sistema vaya mostrando las opciones que contiene con los dígitos ingresados, como cuando se utiliza un buscador web.

PO: ¿Alguna otra información al respecto?

ED: Sí, que el sistema muestre al Conductor la velocidad promedio durante el viaje y la hora de llegada aproximada, actualizando ésta última en función de la velocidad. Y que te permita buscar un destino, a partir de las últimas búsquedas realizadas, por lo menos las últimas 5.

Tema: Revisiones Técnicas al Software

Consigna para los ejercicios de este tema:

- Realice una revisión técnica de código a la siguiente porción de código.
- Complete el reporte de inspección indicado a continuación con los hallazgos identificados.

Reporte de Inspección				
Fecha de Inicio:	<dd/mm/aaaa>			
Revisores:	Nombre	Rol		
Alcance de la revisión:				
Productos de Trabajo Inspeccionados				
Nombre	Versión	Autor		
< Producto de Trabajo Auditado >	<Indicar versión auditada X.X>	<Indicar el autor del producto de trabajo>		
Hallazgos				
Nº	Fecha	Descripción	Línea	Severidad
1				
2				
5				
Decisión Final:		Aceptar/Rechazar/Aceptar Provisoriamente		

Ejercicio 1: Caso Práctico – Sistema GPS

Es el sistema descrito en el ejercicio 4 del tema User Stories.

Presentación de la porción de código

Nota: La clase no está completa, a finalidades del ejercicio tiene sólo 35 líneas.

```
1 //Clase Búsqueda
2 public class Busqueda
3 {
4     private date fechabusqueda;
5     private boolean bandera;
6     Private ArrayList<Destino> Destinos;
7     Public busqueda (Date f)
8     {
9         fechabusqueda = f;
10        Destinos = null;
11    }
12    private boolean AgregarDestinoComoFavorito (Destino Dest)
13    { //inserto el destino "Dest" en la posicion 1 en la lista de favoritos.
14        Boolean agregado = this.Destinos.add(1, Dest);
15        Destino.esFavorito = true;
16        System.out.println("Se agregó el Destino " + Dest.nombre + " como favorito.");
17        return true;
18    }
19 }
20 Public boolean quiarDestinoFavorito (Destino Dest)
21 {
22     String esFavorito = "";
23     Iterator i = Destinos.Iterator();
24     while (i.hasNext()){
25         Destino d = (Destino)i.next();
26         if(d == Dest)
27         {
28             Destinos.remove(d);
29             Dest.esFavorito = false;
30             Return true;
31         }
32         Else
33             return false;
34     }
35 }
36
```

Ejercicio 2: Caso Práctico – Sistema de Afluencia

Presentación de la porción de código

Una empresa de servicios de la Ciudad de Córdoba ha decidido desarrollar un sistema que permita gestionar la afluencia de los clientes, pudiendo además determinar la cantidad de clientes atendidos al final del día y el motivo del contacto. Para este fin se conforma en el área de sistemas de la empresa un Scrum team para que desarrolle el producto, quiénes comenzarán a trabajar en el mismo el primero de agosto.

El **PO** les explica que el sistema a construir deberá permitir a los clientes que se presenten en las oficinas de la empresa emitir un ticket de atención, mediante el cual serán llamados para ser atendidos. Este ticket se imprimirá desde una terminal con pantalla táctil, en donde el cliente deberá seleccionar previamente el tipo de trámite a realizar y, en función de la opción seleccionada, el sistema imprimirá el ticket con una letra y un número.

La letra se asignará acuerdo al trámite a realizar, ya que en función de las mismas se administrarán distintas colas de atención. El número es el orden de atención en la cola de la letra. Por ejemplo: Para el tipo de trámite “Gestión Personalizada”, subtipo “Financiación” se asignará la letra F. Cada día se inicia el contador de atención en cero, con lo cual al primer cliente del día que requiere hacer un cambio de domicilio se le entrega un ticket con el orden F1, al segundo que se presente en el día se le entregará el ticket F2 y así sucesivamente.

Cada uno de los puestos de atención tendrá previamente asignada las letras que atiende, tarea que será realizada por el Supervisor del área (en función de las estadísticas de afluencia por tipo de trámite) y que puede ir cambiando en función de las necesidades del negocio. Por ejemplo, a los Puestos 1 y 2 el día 07/05/2014 se les asigna la letra F. Si hubiera clientes con un ticket con F sin atender, los Representantes de Atención (RA) ubicados en los Puestos 1 y 2 los irán llamando en función del orden de llegada (dados por el número impreso en el ticket)

En cada Puesto el **RA** indicará su estado: “Libre”, “Atendiendo”, “No disponible”. El estado “Atendiendo” se asignará automáticamente cuando desde el Puesto se llama al próximo cliente para ser atendido, el estado Libre se asume cuando se indica el fin de la atención y el estado No disponible es el que se utiliza cuando el RA no está en condiciones de atender clientes. Cuando desde un Puesto de trabajo en estado Libre se llame al siguiente cliente, se informará por 2 pantallas LED el número a ser atendido y el Puesto al que debe dirigirse.

Además de organizar la atención de los clientes en el salón, el sistema deberá poder informar acerca del tiempo promedio de espera de tipo de trámite y el tiempo promedio de atención de los mismos, así como también tiempos máximos y mínimos de espera por rango horario y cantidad de clientes diarios atendidos por rango horario

Presentación de la porción de código

Nota: La clase no está completa, a finalidades del ejercicio tiene sólo 64 líneas.

```

1 //Clase Ticket
2 import java.util.Date;
3 public class Ticket
4 {
5     private int numeroticket;
6     private TipoTramite tipoTramite;
7     private SubtipoTramite subtipoTramite;
8     private Date fechaEmision;
9     private char letra;
10    private Date tiempoEspera;
11
12    public Ticket()
13    {
14    }
15
16    public String conocerLetra(tipoTramite,SubtipoTramite)
17    {
18        //Falta resolver la asignación de la cola de atención según el tipo
19        //y subtipo de trámite
20        .....
21    }
22
23    public string conocerTipoTramite()
24    {
25        return TipoTramite.mostrarTipoTramite();
26    }
27
28    public String conocersubtipoTramite()
29    {
30        return subtipoTramite.mostrarTipoTramite();
31    }
32
33    }

```

```

35 //Clase TipoTramite
36
37 public class TipoTramite
38 {
39     private int ID;
40     private String descripcion;
41
42     public TipoTramite()
43     {
44     .....
45     }
46     public String conocerTipoTramite()
47     {
48         return System.out.println(descripcion);
49     }

```

```

50
51 //Clase subtipoTramite
52 public class subtipoTramite
53 {
54     private String descripcion;
55
56     public subtipoTramite()
57     {
58     }
59
60     public String conocerSubtipoTramite()
61     {
62         //public String getDescripcion()
63     }
64 }

```

Ejercicio 3: Caso Práctico – Empresa de Televisión por Cable

El área de desarrollo de sistemas de una empresa de televisión por cable se encarga de realizar desarrollos para los sistemas que dan soporte a la prestación de servicios de la empresa. En este momento se les ha presentado el desafío de realizar un desarrollo innovador, para brindar a los usuarios del servicio de cable una interfaz de consulta de programación televisiva ágil y con diferentes formas de búsqueda, entre las que se incluye: búsqueda por tipo de programación (películas, series, informativos, deportes, etc.), búsqueda por nombre de la programación, búsqueda por elenco, búsqueda por tipo de visualización (HD o Analógica), etc. Al tratarse de un proyecto de innovación, ésta área ha definido trabajar con un framework Scrum, donde el Gerente de Desarrollo fue designado como el Product Owner, debido a que cuenta con mucha experiencia en la empresa, conoce los servicios y el funcionamiento actual de los sistemas. A esto se le suma que la idea original del proyecto fue suya.

Presentación de la porción de código:

Nota: La clase no está completa, a finalidades del ejercicio tiene sólo 20 líneas.

```
1 class Busq{
2   // @autor CM
3   // private List <CriterioBusqueda> criteriosBusqueda;
4   private boolean hd;
5   public List <TipoProgramacion> tipos;
6
7   // @Method Constructor por defecto
8   public Busq() {
9       // criteriosBusq = new List();
10      hd = false;
11      tipos = null; }
12
13   private TipoProgramacion busquedaPortipoProgramacionyVisualizacion(boolean
14   visualizacion; Tipoprogramación tipo) {
15       Iterator i = tipos.Iterator();
16       while (i.hasNext()){ TipoProgramacion tipoActual = (TipoProgramacion)i.next();
17       if ( hd && tipoActual == tipo) return tipo;
18       else { if (!hd&& tipos.getValue(i)==tipo) return tipo;}}
19   return null;}
20 }
```

Ejercicio 4: Caso Práctico – Desarrollos a Medida

Presentación de la porción de código

Nota: La clase no está completa, a finalidades del ejercicio tiene sólo 32 líneas.

```
1  import java.util.ArrayList;
2  //Clase Localidad
3  public class Localidad {
4      private String DESCRIPCION;
5      private String nombre;
6      private ArrayList<Barrio> barrios;
7
8      public Localidad(String nomb) {
9          nombre = nomb;
10     }
11
12     public String getNombre() {
13         return nombre;
14     }
15     public void setNombre(String nombre) {
16         this.nombre = nombre;
17     }
18
19     public Barrio[] getBarrios() {
20         return barrios;
21     }
22
23     public void addBarrio(Barrio barrio) {
24         int banderaA = 0;
25         if(this.barrios == null) {
26             this.barrios = new Barrio();
27         }
28         boolean banderaB = this.barrios.add(barrio);
29
30         return bandera;
31     }
32 }
```


Tema: Administración de Configuración de Software

Consigna para los ejercicios de este tema:

- Realizar una propuesta de estructura del repositorio para el producto y para el proyecto.
- Identificar al menos 5 Ítems de Configuración del proyecto y del producto, indicando para cada uno: Nombre, Regla de Nombrado y Ubicación dentro del repositorio. Distinguir entre ítems del producto e ítems del proyecto y, para estos últimos, diferenciar aquellos comunes a todo el proyecto y los propios de cada sprint.
- Utilizar el formato siguiente:

Nombre del Ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de Ítem [Producto / Proyecto / Iteración]

Ejercicio 1: Caso Práctico – GPS

Presentación del Caso de Estudio

Consideraciones para el Proyecto

Para construir el software, la empresa desarrolladora utilizará un proceso de desarrollo ágil, específicamente con un framework SCRUM. El equipo de proyecto trabaja de la siguiente manera: Cuando la información no es clara o está incompleta, como en este caso, el Product Owner mantiene entrevistas con los expertos del dominio, para luego realizar reuniones con el equipo del proyecto para definir los requerimientos, los cuales se registran en un sistema como tarjetas que representan user stories.

Cuando se realizan las reuniones de planificación (planning meeting), en el pizarrón se colocan post-its que identifican cada actividad a realizar, el responsable y la estimación correspondiente a la tarea. Una vez armado el pizarrón se le saca una fotografía para contar con la información correspondiente al inicio de un sprint.

Durante la ejecución del sprint al realizar las daily meeting (reuniones diarias) en un pizarrón se genera el burndown chart para ir conociendo el avance del trabajo y se genera, si es necesario la lista de impedimentos y se gestiona la lista de riesgos identificados. Luego en las reuniones de revisión del sprint (sprint review meetings) se compara el gráfico burndown chart con la capacidad definida para el sprint.

Cabe aclarar que el área cuenta con la herramienta Clear Case, por lo que el versionado de todo el código fuente y documentación necesaria se realiza con esta herramienta. Se está trabajando con la modalidad de vistas dinámicas, por lo que siempre se cuenta con una vista actualizada de toda la información.

Para generar el código se cuenta con una herramienta con interfaz gráfica de desarrollo sincronizada con Clear Case, de tal forma de ir manteniendo el versionado del código fuente en el repositorio del producto. A su vez, la herramienta cuenta con opciones para compartir el escritorio, lo cual facilita la ejecución de revisiones de pares del código generado. Estas revisiones se ejecutan y los hallazgos identificados por el revisor son resguardados para luego realizar un seguimiento y verificar que se corrijan dichos hallazgos.

Ejercicio 2: Caso Práctico – Herramienta de Seguimiento de Defectos

Presentación del Caso de Estudio

Una empresa se dedica al desarrollo y mantenimiento de software a medida para distintos clientes. Los productos de software que se construyen son basados en Windows. El motivo por el cual se requiere el producto, es porque luego de haber implementado varios sistemas para distintos clientes, han detectado que los productos de software que entregan presentan defectos que no son encontrados en las etapas de desarrollo.

Esto hace que sus clientes tengan que registrar los errores en papel, enviar un mail, o bien, hacer un llamado telefónico para informar los detalles del defecto encontrado al equipo de desarrollo.

Haciendo un análisis hacia adentro, también han detectado que los equipos de desarrollo de la empresa no cuentan con un proceso de registro, evaluación, asignación y seguimiento de los defectos que se detectan en las etapas de desarrollo.

Es por esto que han decidido implementar un sistema de seguimiento de defectos (*Bug Tracking System*) sencillo que se adapte a sus necesidades puntuales.

Con este sistema se pretende:

- Establecer un proceso e implementarlo a través de una herramienta que permita gestionar los defectos encontrados de manera de tratar de reducir la cantidad de defectos que son detectados por los usuarios finales
- En el caso de que el usuario final detecte un defecto, contar con un mecanismo simple y uniforme para que el usuario pueda informar la existencia de los mismos.

El mismo sistema debería permitir registrar los defectos de todas las aplicaciones que están en desarrollo, aplicaciones instaladas en los distintos clientes y nuevas aplicaciones a desarrollar.

Desde ya que cada usuario (cliente o desarrollador) solo debe poder operar con la información de defectos correspondientes a las aplicaciones con las que está asociado.

Debido a que no solo los desarrolladores sino también los clientes de la empresa deben poder acceder al sistema, se necesita que el mismo esté disponible a través de Internet. Dicha disponibilidad debería ser permanente durante el horario de trabajo de la empresa, como así también durante el horario de trabajo de sus clientes.

Otro punto importante a considerar es que el manejo del sistema debe ser simple, para que cualquier persona con conocimientos básicos de manejo de PC lo pueda utilizar; y rápido, principalmente en el momento de registrar los defectos.

Con la intención de obtener a corto plazo una versión funcional del producto, que luego se irá extendiendo y mejorando a partir de la validación con sus clientes, se ha decidido emplear un ciclo de vida Iterativo e Incremental para este proyecto, definiendo tres iteraciones para completar el producto.

El proyecto tiene previsto realizar entrevistas para relevar los requerimientos de los distintos clientes; las que podrán repetirse en cada iteración según necesidad, generando la actualización de la Especificación de Requerimientos de Software del producto (ERS). Todas las entrevistas deberán estar documentadas en minutas de relevamiento.

El proyecto incluye despliegue del sistema, su parametrización y puesta a punto para comenzar a trabajar, que incluye la carga inicial y la parametrización. La carga de datos inicial es mucha y para la época en que

se realicen esas actividades está previsto que el personal se tome vacaciones, ya que sería a fines de diciembre o primeros días de enero. Las personas relevadas hasta el momento han puesto muchas expectativas sobre los beneficios del sistema y son todos muy sensibles al aspecto estético que tengan las pantallas y los reportes, ya que casi todos los empleados no son especialistas en el manejo de software.

Se está evaluando utilizar como lenguaje de programación VB.Net, pero se requerirá una capacitación especial para los programadores y algunos otros miembros del equipo. El presupuesto inicial que se pasó por el desarrollo no tenía prevista la contratación de terceros, por lo que el Líder de Proyecto deberá estar muy atento para no exceder su presupuesto. Se empleará la herramienta Subversion para el control de versiones de los ítems de configuración y líneas base del proyecto.

Es necesario dimensionar los recursos necesarios para las pruebas, ya que no se sabe si el hardware disponible para pruebas será el necesario y suficiente para probar el producto en la etapa de prueba de sistemas.

Como resultado de la Planificación, se generarán los siguientes documentos: Estimación, WBS, Cronograma del Proyecto (Gantt), Plan de Desarrollo de Software, Plan de Iteración y Planes de Soporte (Plan de Gestión de Riesgos, Plan de Calidad, Plan de Prueba del Proyecto, Plan de Despliegue, Plan de Aceptación del Producto, Plan de Gestión de Configuración de Software, Plan de Participación de Involucrados).

La Gerencia ha solicitado que cada 15 días el Líder de Proyecto informe sobre el avance del proyecto, para lo cual deberá generar un Reporte de Control de Estado.

Las reuniones de seguimiento del proyecto deberán estar documentadas bajo las respectivas minutas de reunión, dejando constancia del tema tratado, decisiones tomadas, fecha y hora de inicio, participantes y duración de la misma.

El plazo máximo estipulado para que el producto esté funcionando y en régimen es de 6 meses.

Ejercicio 3: Caso Práctico – Liga Regional de Fútbol

Presentación del Caso de Estudio

La Asociación Cordobesa de Fútbol ha contratado a una empresa que se dedica al desarrollo de software a medida para distintos clientes, con el objetivo de contar con un producto de software que le ayude con la administración de los campeonatos (y los clubes que participan en los mismos) que organiza a través de las Ligas Regionales de Fútbol.

Todos los años se realizan campeonatos de fútbol. Los clubes que desean participar pueden inscribirse en la Liga hasta un mes antes del comienzo del campeonato.

Luego de la inscripción cada club debe presentar la lista de sus jugadores, un examen médico para cada uno, y los datos de la cancha habilitada para los partidos en los que participe como local. Cuando se presenta dicha lista el jugador queda registrado para participar con ese club en el campeonato. No obstante, hasta que no presente el examen médico no estará habilitado para jugar.

Cuando la Liga realiza la diagramación de un campeonato, de acuerdo al número de clubes inscriptos, determina la cantidad de jornadas o “fechas” a realizarse, y para cada una se definen los partidos a jugarse. Además, para cada partido se indica la cancha en que se realizará, los árbitros que intervendrán y el rol de cada uno en ese partido (árbitro principal, juez de línea, etc.).

Luego de culminada cada fecha, deben registrarse los resultados de cada partido:

- Goles convertidos: Indicando para cada gol el jugador que lo realizó y el tiempo de juego transcurrido.
- Amonestaciones efectuadas: Indicando para cada una el tipo de tarjeta (roja o amarilla), el jugador y el tiempo de juego transcurrido.

La Asociación Cordobesa de Fútbol desea tener un registro para cada jugador del tiempo jugado en cada partido y asignarle una calificación en cada partido, para luego poder premiar o reconocer de alguna manera a aquellos con el promedio más alto y la mayor cantidad de minutos jugados en un campeonato.

Además, se desea emitir una lista de goleadores (jugadores que convirtieron más goles) para premiar al goleador de cada campeonato. Para todos los reportes y estadísticas se ha requerido que los mismos se muestren tanto en formato de tablas como en formato gráfico. Todos los involucrados han puesto muchas expectativas sobre los beneficios del sistema y son todos muy sensibles al aspecto estético que tengan las pantallas y los reportes, ya que la mayoría de los empleados no son especialistas en el manejo de software.

Para realizar el desarrollo del sistema, la empresa desarrolladora utilizará un proceso de desarrollo tradicional. Con la intención de obtener a corto plazo una versión funcional del producto, que luego se irá extendiendo y mejorando, se ha decidido emplear un ciclo de vida Iterativo e Incremental para este proyecto, definiendo tres iteraciones para completar el producto.

El alcance del proyecto implica la especificación de los requerimientos, el desarrollo del producto, las pruebas, el despliegue del mismo y la capacitación a un representante de cada una de las Ligas Regionales. Es fundamental para el éxito del proyecto la aceptación del producto no sólo por parte de los referentes de cada Liga Regional sino también por los usuarios de cada uno de los clubes, dado que se ha decidido que el software permita que cada club pueda gestionar un usuario y una clave de acceso y pueda registrar su propia inscripción y la inscripción de la lista de jugadores que participará. Como consecuencia, el producto de software no sólo deberá desarrollarse con tecnología web, sino que el manejo del sistema debe ser simple para que cualquier persona con conocimientos básicos de manejo de PC lo pueda utilizar; y rápido, principalmente en el momento de registrar los datos de la inscripción.

El plazo expuesto para la finalización del proyecto es de 40 semanas y se requiere que además se contemple un período de “garantía” de 15 días hábiles. En el contrato están estipuladas multas que la empresa deberá pagar por incumplimiento de plazos. Además, el presupuesto inicial que se pasó por el desarrollo no tenía prevista la contratación de terceros. Por lo que el Líder de Proyecto deberá estar muy atento para no excederse en tiempos y en el presupuesto.

Se está evaluando utilizar como lenguaje de programación VB.Net, pero se requerirá una capacitación especial para los programadores y algunos otros miembros del equipo. Se emplearán las herramientas Subversion y TortoiseSVN para el control de versiones de los ítems de configuración y líneas base del proyecto.

El proyecto tiene previsto realizar entrevistas para relevar los requerimientos; las que podrán repetirse en cada iteración según la necesidad, generando la actualización de la Especificación de Requerimientos de Software del producto (ERS). Todas las entrevistas deberán estar documentadas en minutas de relevamiento.

Como resultado de la Planificación, se generarán los siguientes documentos: Estimación, WBS, Cronograma del Proyecto (Gantt), Plan de Desarrollo de Software, Plan de Iteración y Planes de Soporte (Plan de Gestión de Riesgos, Plan de Calidad, Plan de Prueba del Proyecto, Plan de Despliegue, Plan de Aceptación del Producto, Plan de Gestión de Configuración de Software y Plan de Participación de Involucrados).

La Asociación Cordobesa de Fútbol ha solicitado que cada 15 días el Líder de Proyecto informe sobre el avance del proyecto, para lo cual deberá generar un Reporte de Control de Estado.

Las reuniones de seguimiento del proyecto deberán estar documentadas bajo las respectivas minutas de reunión, dejando constancia del tema tratado, decisiones tomadas, fecha y hora de inicio, participantes y duración de la misma. Estas minutas podrán ser enviadas por mail a todo el equipo del proyecto, por lo que resulta importante resguardar estos mails.

Ejercicio 4: Caso Práctico – Software para Call Center²

Presentación del Caso de Estudio

Con motivo de la modernización de su Call Center, una empresa de servicios ha solicitado a una consultora de desarrollo de software la construcción de un producto destinado a facilitar la gestión de las llamadas, desde que el cliente se comunica con la empresa hasta que la llamada es finalizada. Con este nuevo producto la empresa espera además obtener mediciones que le permitan mejorar el servicio.

Para este fin la consultora conforma un Scrum team para que desarrolle el producto, quiénes comenzarán a trabajar en el mismo el primero de agosto. En la grooming meeting el Product Owner (PO) le plantea al equipo que para enero del año próximo (aprovechando la menor afluencia de llamadas por el período vacacional) está previsto un importante corte del servicio por razones de mantenimiento, con lo cual para ese momento el sistema debería estar disponible para ser utilizado, mejorando así la atención a los clientes que llamen para informarse.

A continuación, se transcribe parte de la entrevista entre el Product Owner (PO) y el Scrum Team (ST):

PO: El software a construir deberá conectarse al sistema telefónico de líneas rotativas, recibir las llamadas de los clientes y derivarlos a los Operadores Telefónicos para su atención, según corresponda. Para esto deberá contar con una herramienta para generación de campañas de IVR que permita construir distintos circuitos de atención mediante la generación de “árboles de decisión y/o procesos” en forma gráfica. Para esto necesitamos una interfaz GUI (Interfaz de Usuario Grafica), que pueda ser utilizada por el Supervisor del Call Center.

ST: ¿Qué es un IVR?

PO: La respuesta de voz interactiva o IVR (del inglés Interactive Voice Response) consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como «sí», «no» u otras. Es un sistema automatizado de respuesta interactiva, orientado a entregar o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras operaciones.

ST: ¿Cómo funciona?

PO: El cliente realiza una llamada a un número de teléfono, el sistema de audio contesta la llamada y le presenta al cliente una serie de acciones a realizar, esto se hace mediante mensajes (menús de opciones) previamente grabados en archivos de audio (por ejemplo, «pulse o diga uno para ventas, dos para administración»). El cliente elige la opción a realizar introduciendo un número en el teclado del teléfono o diciendo dicho número y navega por los diferentes menús hasta encontrar la información solicitada o que el sistema dirija la llamada al destinatario elegido.

ST: ¿Qué significa que el sistema permita generar campañas de IVR?

² **Call center** es una expresión de la lengua inglesa que puede traducirse como centro de llamadas. Se trata de la oficina donde un grupo de personas específicamente entrenadas se encarga de brindar algún tipo de atención o servicio telefónico.

PO: Quiere decir que las opciones que el sistema le presenta al cliente a través del IVR son configurables. Por ejemplo, ante un corte de servicio puede informarle al cliente, previa solicitud del ingreso de su número de cliente, si está afectado o no con dicho corte. Una vez finalizado el corte esa rama del árbol del IVR se deshabilita.

ST: ¿Y cuándo interviene el Operador Telefónico?

PO: Cuando el cliente no puede resolver su consulta a través del IVR es derivado a una Cola de atención, en donde lo atenderá un Operador Telefónico, que tiene asignado un número de posición.

ST: ¿Le sirve de utilidad la información solicitada por el IVR al Operador Telefónico?

PO: Sí. Si se le solicita al cliente el motivo de su llamada el Operador Telefónico contará con esta información al momento de atenderlo.

ST: ¿Cómo distribuye el sistema las llamadas entre los operadores?

PO: El sistema deberá distribuir las llamadas entrantes en cola a las distintas posiciones de Operadores Telefónicos, permitiendo una carga de trabajo uniforme por operador. Esta distribución se realizará en base a grupos de atención, especializados por temas, tipos de cliente, etc., de acuerdo al perfil del Operador Telefónico (Skill). Esta distribución podrá ser cambiada dinámicamente por el Supervisor, siguiendo la evolución de los requerimientos sobre el Call Center y de acuerdo a las habilidades y conocimientos de los Operadores Telefónicos.

ST: ¿Cómo conoce el sistema que operadores telefónicos están disponibles para recibir llamadas?

PO: El sistema deberá administrar los estados del Operador Telefónico agrupados bajo la siguiente disposición:

- **Listo:** este estado se utiliza cuando el Operador Telefónico desea recibir llamadas provenientes de las colas. Optar por este estado implica conectarse a la/s cola/s que cada Operador Telefónico tenga configuradas, en base al grupo al cual pertenece y al perfil configurado por el supervisor.
- **Ocupado:** una vez que el Operador Telefónico recibe una llamada de una cola pasa automáticamente al estado Ocupado. Este estado indica que el Operador Telefónico se encuentra cursando una llamada, por lo que el sistema no cuenta con él para la transferencia de las próximas llamadas mientras se encuentre en ese estado. Se trata de un estado totalmente forzoso y automático, dado que es el sistema y no el Operador Telefónico es quien cambiará el estado de Listo a Ocupado al transferirle una llamada. En este estado el supervisor tendrá la posibilidad de monitorear la llamada en curso.
- **Libre:** este estado indica que el Operador Telefónico se encuentra dentro del ámbito de trabajo (logueado) y que no se encuentra hablando, pero que no desea recibir llamadas. Dicho estado no es forzoso, sino que es el Operador Telefónico quien decide cuando usarlo. Es útil para denotar tareas que no tengan que ver con la gestión telefónica, como la carga de datos, el envío de información por e-mail, entre otras. Equivale al deslogueo de las colas, por lo que al entrar en este estado el sistema deja de considerar a este Operador Telefónico como parte del grupo disponible.
- **Fuera:** indica que el Operador Telefónico no se ha logueado en la Plataforma, probablemente porque no se encuentra en su ámbito de trabajo. En este estado no se realiza ningún tipo de cálculo de tiempos del Operador Telefónico, pues este no se encuentra trabajando.
- **Pausas configurables:** esta funcionalidad es de suma utilidad a la hora de intentar medir cuanto tiempo le dedican los Operador Telefónicos a ciertas tareas que no tienen que ver con la gestión telefónica, ya sean

parte o no del trabajo del Operador Telefónico. Ejemplos de configuración de estas pausas pueden ser: Descanso, Almuerzo, Baño, Capacitación, Consulta al supervisor o Tareas administrativas.

ST: ¿Qué ocurre si el Operador Telefónico al que el sistema deriva la llamada no atiende?

PO: En el sistema deberá ser posible determinar el tiempo máximo de espera (cantidad de Rings) en ser atendido cuando es pasada la llamada a un Operador Telefónico y no es atendida. En caso de superarse ese tiempo, la llamada es reasignada automáticamente a otro Operador Telefónico y le cambia el estado a Libre al Operador Telefónico que no respondió.

ST: ¿Cómo monitorea el supervisor la operatoria del Call Center?

PO: El sistema deberá permitir monitorear tanto los distintos estados de un Operador Telefónico como el de todo el sistema en base a reportes en tiempo real y monitoreo en línea (semáforo), similar al que se muestra a continuación:



El sistema también deberá proveer estadísticos y reportes históricos, para un análisis posterior.

El equipo estableció que los Sprints sean de una duración de 15 días. Se considera que el éxito del sistema a construir tiene un impacto directo en la imagen de la empresa de servicios, ya que afecta directamente la calidad de atención. Asimismo, es de suma importancia asegurar la calidad desde la primera versión del producto, ya que su buen o mal funcionamiento impacta directamente en el cliente al momento de comunicarse con la empresa.

Con el objetivo de contar con diferentes configuraciones para cada entorno de trabajo (desarrollo, testing, preproducción) se guardarán los diferentes valores para cada servicio como conexión a la base de datos, configuración del servidor de email saliente, motor de caché, etc. en diferentes carpetas. Además, para mantener la sincronización entre las bases de datos de los desarrolladores el equipo ha implementado una estrategia llamada “migraciones”, con la cual se crean diferentes archivos de script incrementales ordenados por fecha que van incorporando los cambios en las tablas afectadas.

Soluciones Propuestas

Tema: User Stories

Ejercicio 1: Caso Práctico – Festival de Folklore

Presentación del Caso de Estudio

Anualmente la Dirección de Cultura de la Municipalidad de una localidad de la provincia, organiza un festival de folklore. Este festival tiene una duración de generalmente cinco noches, aunque esto puede variar de año en año. En cada una de las noches actúan distintos grupos folklóricos con reconocimiento regional, provincial y nacional. El festival se prepara con mucha anticipación y se realiza la diagramación para determinar qué grupos actúan en cada noche y el orden en el que los mismos realizarán sus presentaciones, teniendo en cuenta que los horarios de presentación de los grupos no pueden superponerse y que no pueden quedar espacios sin ninguna presentación entre medio de dos grupos. Considerar que no puede incluirse la participación de un grupo más de una vez para un mismo festival, en una misma noche.

En cada noche se define la hora de inicio de la misma, pero no se determina la hora de fin, ya que esta puede variar según si las presentaciones se extienden más de lo previsto.

El Festival se realiza en un único estadio, que está dividido en sectores (A, B, C, etc.), que se identifican con colores diferentes, y cada sector se compone de filas (1, 2, 3, etc.), cada fila, a su vez, está conformada por butacas, las cuales están numeradas.

La venta de entradas se realiza en cinco puntos de venta que se encuentran en funcionamiento simultáneamente: en el estadio donde se realizará el festival, en tres centros comerciales de la ciudad capital y en un centro comercial de la localidad dónde se realiza el festival. No se debe permitir que se venda una misma entrada (una misma butaca de un festival en una misma fecha) en dos puntos de venta diferentes.

Existen distintos tipos de entradas para el público (mayores, menores, jubilados, etc.). El precio de las entradas depende del tipo de entrada y del sector donde se encuentre la butaca, además puede variar de una noche a otra, dependiendo de los grupos musicales que actúan. Por ejemplo, una entrada para mayores en el sector A, que está cerca del escenario, será más costosa que una para mayores en el sector E que está más alejado del mismo y a su vez puede variar de noche en noche el precio de la entrada en la misma ubicación. Las butacas se venden para una noche en particular así es que una misma butaca puede estar disponible, por ejemplo, para la noche 1 y 3, y ocupada para la noche 2, 4 y 5.

También se habilita la venta anticipada de las entradas a un precio menor, un porcentaje de descuento que la Dirección de Cultura determina, al igual que la fecha de vencimiento de ese beneficio, por ejemplo, venta anticipada con un descuento del 10 % hasta un mes antes que empiece el festival. La forma de venta de entradas es únicamente de contado en efectivo. Si un cliente solicita la anulación de la entrada sólo se le reintegra el 50% del monto abonado. Esto se puede hacer hasta 10 días antes del inicio del festival.

La entrada tiene un código de barras para evitar falsificaciones. Además, hay que tener en cuenta que la misma entrada cumple la función de factura, por lo que debe tener los datos requeridos por la ley de facturación, y debe asegurarse de que el número de factura sea único.

La Dirección de Cultura de la Municipalidad ha solicitado a su Área de Sistemas el desarrollo de un sistema de información que le ayude con la administración de los festivales que organiza, la diagramación de la programación y la venta de entradas y brinde información que ayude a la organización de próximos

festivales. La Dirección de Cultura de la Municipalidad tiene licencias para realizar la aplicación con una base de datos Oracle.

Debido a que en las horas pico se suele generar cola en los puntos de venta, es necesario que el sistema genere una entrada en no más de 6 segundos.

Solución Propuesta

Roles

- ↔ **Vendedor de Entradas**
- ↔ **Director de Cultura de la Municipalidad**
- ↔ **Responsable de Festival**
- ↔ **Responsable de Predio**

Nota: las user stories resaltadas en verde son resultado de la división de la user story inmediata anterior resaltada en amarillo.

Vender Entradas	5
<p>Como Vendedor de Entradas quiero poder vender entradas para un festival vigente para cumplir con mis objetivos de venta.</p> <p>Notas:</p> <ul style="list-style-type: none"> • Sólo se acepta efectivo. • Tiempo de respuesta no mayor a 6 segundos. • Debe imprimir un código de barras. • Debe controlar concurrencia de la butaca a vender 	
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar vender una entrada. (pasa) <input type="checkbox"/> Probar vender más de una entrada del mismo tipo de entrada. (pasa) <input type="checkbox"/> Probar vender más de una entrada de diferentes tipos de entrada. (pasa) <input type="checkbox"/> Probar vender entradas de un único sector. (pasa) <input type="checkbox"/> Probar vender entradas de más de un sector. (pasa) <input type="checkbox"/> Probar con descuento por venta anticipada. (pasa) <input type="checkbox"/> Probar con descuento por venta anticipada, cuando no corresponde. (falla) <input type="checkbox"/> Probar sin descuento por venta anticipada. (pasa) <input type="checkbox"/> Probar vender ubicaciones tomadas por otro punto de venta.(falla) <input type="checkbox"/> Probar imprimir entradas (pasa) <input type="checkbox"/> Probar imprimir entradas con impresor apagado (falla) 	

<p>Generar Entradas</p> <p>Como Vendedor de Entradas quiero poder generar entradas para un festival vigente para cumplir con mis objetivos de venta.</p> <p>Notas:</p> <ul style="list-style-type: none"> • Sólo se acepta efectivo. • Debe controlar concurrencia de la butaca a vender 	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar vender una entrada. (pasa) <input type="checkbox"/> Probar vender más de una entrada del mismo tipo de entrada. (pasa) <input type="checkbox"/> Probar vender más de una entrada de diferentes tipos de entrada. (pasa) <input type="checkbox"/> Probar vender entradas de un único sector. (pasa) <input type="checkbox"/> Probar vender entradas de más de un sector. (pasa) <input type="checkbox"/> Probar con descuento por venta anticipada. (pasa) <input type="checkbox"/> Probar con descuento por venta anticipada, cuando no corresponde. (falla) <input type="checkbox"/> Probar sin descuento por venta anticipada. (pasa) <input type="checkbox"/> Probar vender ubicaciones tomadas por otro punto de venta.(falla) 	
<p>Imprimir Entrada</p> <p>Como Vendedor de Entradas quiero poder imprimir entradas para un festival vigente para cumplir con mis objetivos de venta.</p> <p>Notas:</p> <ul style="list-style-type: none"> • Tiempo de respuesta no mayor a 6 segundos. • Debe imprimir un código de barras. 	2
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar imprimir una entrada (pasa) <input type="checkbox"/> Probar imprimir 5 entradas (pasa) <input type="checkbox"/> Probar imprimir entradas con el impresor apagado (falla) 	
<p>Registrar festival</p> <p>Como Responsable de Festival quiero diagramar un festival para organizar su realización.</p> <p>Nota:</p> <ul style="list-style-type: none"> • No puede haber festivales simultáneos 	1
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar crear festival con todos los datos necesarios. (pasa) <input type="checkbox"/> Probar crear un festival que tenga superposición de fechas con uno existente. (falla) <input type="checkbox"/> Probar crear un festival con datos faltantes (falla) 	

<p>Diagrama Festival:</p> <p>Como Responsable de Festival quiero diagramar un festival para definir qué grupos musicales actuarán cada día.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Tener en cuenta restricciones de diagramación. • Poder dejar una diagramación incompleta 	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar crear diagramación que cumpla con las restricciones. (pasa) <input type="checkbox"/> Probar crear diagramación que NO cumpla las restricciones. (falla) <input type="checkbox"/> Probar crear diagramación incompleta. (pasa) 	
<p>Administrar distribución del estadio</p> <p>Como Responsable de Predio quiero registrar el estadio para tener el predio donde se realizarán los festivales</p> <p>Nota:</p> <ul style="list-style-type: none"> • Los sectores del estadio deben identificarse con colores diferentes. • Cada sector incluye un conjunto de filas que incluye varias butacas. 	2
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar crear un estadio con varias butacas, filas y columnas. (pasa) <input type="checkbox"/> Probar agregar una fila a un sector existente. (pasa) <input type="checkbox"/> Probar incluir una butaca en dos filas distintas. (falla) <input type="checkbox"/> Probar incluir varios sectores de diferentes colores cada sector. (pasa) <input type="checkbox"/> Probar incluir varios sectores del mismo color. (falla) 	
<p>Registrar grupo musical</p> <p>Como Responsable del Festival quiero registrar un grupo musical para incluirlo en la diagramación del uno o más festivales</p> <p>Nota:</p> <ul style="list-style-type: none"> • Un grupo musical puede tener uno o más artistas • Un mismo artista puede incluirse en más de un grupo musical 	3
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar crear un grupo musical con un único artista. (pasa) <input type="checkbox"/> Probar crear un grupo musical con varios artistas. (pasa) <input type="checkbox"/> Probar completar todos los datos de algunos artistas y no completar otros datos de un artista. (falla) <input type="checkbox"/> Probar crear un grupo musical con un nombre previamente existente (falla) 	

<p>Registrar grupo musical solista</p> <p>Como Responsable del Festival quiero registrar un grupo musical de un artista para utilizarlo en la diagramación del Festival</p>	1
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar crear un grupo musical con un único artista. (pasa) <input type="checkbox"/> Probar crear un grupo musical con un único artista sin completar todos los datos. (falla) <input type="checkbox"/> Probar crear un grupo musical con un nombre previamente existente (falla) 	
<p>Registrar grupo musical con más de un artista</p> <p>Como Responsable del Festival quiero registrar un grupo musical integrado por dos o más artistas para utilizarlo en la diagramación del Festival</p>	2
<p>Nota: Un mismo artista puede incluirse en más de un grupo musical</p> <p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar crear un grupo musical con varios artistas. (pasa) <input type="checkbox"/> Probar completar todos los datos de algunos artistas y no completar otros datos de un artista. (falla) <input type="checkbox"/> Probar crear un grupo musical con un nombre previamente existente (falla) <input type="checkbox"/> Probar incluir el mismo artista en distintos grupos musicales (pasa) 	
<p>Definir precios</p> <p>Como Responsable de Festival quiero fijar precios para un festival</p>	2
<p>Nota: tener en cuenta que el precio depende de la noche del festival, el sector y el tipo de entrada</p> <p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar registrar precios para una noche, sector y tipo de entrada que ya los tiene (falla) <input type="checkbox"/> Probar registrar precios para una noche, sector y tipo de entrada (pasa) <input type="checkbox"/> Probar registrar precios para un festival vigente (pasa) <input type="checkbox"/> Probar registrar precios cuando no hay festival vigente (falla) 	

<p>Habilitación de butacas para venta: Como Responsable de Predio quiero definir qué butacas se habilitar para la venta de entradas</p>	2
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar habilitar todos los sectores con sus butacas. (pasa) <input type="checkbox"/> Probar habilitar algunas butacas de un sector. (pasa) <input type="checkbox"/> Probar no habilitar ninguna butaca. (falla) 	
<p>Registrar punto de venta Como Responsable de Festival quiero registrar un punto de venta para realizar la venta de entradas desde estos puntos de venta</p>	1
<p>Nota: Un punto de venta forma parte de un centro de venta</p> <p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar registrar un punto de venta con todos sus datos. (pasa) <input type="checkbox"/> Probar registrar un punto de venta con algunos datos faltantes. (falla) <input type="checkbox"/> Probar registrar un punto de venta sin seleccionar un centro de venta al que pertenece. (falla) 	
<p>Reporte de asistencia de público Como Director de Cultura de la Municipalidda quiero obtener información sobre el público que asiste a los festivales para conocer la aceptación del festival en cuanto a concurrencia de público</p>	3
<p>Nota: Incluir gráficos de torta que muestren el porcentaje de ocupación del estadio</p> <p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar generar un reporte con el 100% del estadio ocupado. (pasa) <input type="checkbox"/> Probar generar un reporte con el 0% del estado ocupado. (pasa) <input type="checkbox"/> Probar generar un reporte sin seleccionar todos los datos para generarlo. (falla) <input type="checkbox"/> Probar generar un reporte con más del 0% y menos del 100% del estadio ocupado (pasa) 	

<p>Reporte de venta de entradas</p> <p>Como Director de Cultura de la Municipalidad quiero obtener información sobre la venta de entradas de un festival para analizar que tan bien diagramado estuvo y cómo resultó la selección de artistas de cada día.</p> <p>Nota: Incluir los totales por festival y por noche de festival</p>	3
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar generar un reporte con el 100% de la venta. (pasa) <input type="checkbox"/> Probar generar un reporte con el 0% de la venta. (pasa) <input type="checkbox"/> Probar generar un reporte sin seleccionar todos los datos para generarlo. (falla) <input type="checkbox"/> Probar generar un reporte con más del 0% y menos del 100% de la venta (pasa) <input type="checkbox"/> Probar generar un reporte con el 0% de venta para una noche, entre el 1% y el 99% para otra noche, y el 100% para otra noche. (pasa) 	

Restricciones (Requerimientos no funcionales)

<p>La base de datos a utilizar debe ser Oracle</p> <p>Restricción</p>	0
<p>El sistema debe soportar el uso de hasta 20 usuarios concurrentes</p> <p>Restricción</p>	0

Tema: User Stories

Ejercicio 2: Caso Práctico – Taxi Mobile: Sistema web mobile para seguimiento de taxis

Presentación del Caso de Estudio

Objetivo: Desarrollar un producto de software para Smart phones, donde los pasajeros de taxis puedan solicitar el taxi más cercano y saber su ubicación y demora en todo momento.

A **continuación**, se transcribe parte de la entrevista realizada al experto en el dominio:

Product Owner (PO): ¿La idea es que el pasajero pueda llamar a una central sabiendo dónde está el taxi que va a pedir? ¿Cómo pide un taxi el pasajero?

Experto en el Dominio (ED): El pasajero debe ingresar a la aplicación, la cual activará el sistema de posicionamiento. El sistema detecta dónde está el pasajero y le muestra los 5 taxis más próximos a su ubicación, visualizados en un mapa, e informa ubicación, distancia y tiempo estimado. El pasajero selecciona el taxi deseado y esta acción envía una notificación a la central de taxis y al taxista a su celular. De esta forma ambos identifican el pedido de un móvil, y el taxista puede saber a dónde está el pasajero

PO: ¿Y cómo sabe el taxista quién es el pasajero?

ED: Cuando se instala la aplicación en el celular se pedirán los datos mínimos para registrarlo como usuario. Estos datos se pueden obtener de Facebook y el número de celular del mismo aparato donde se instala la aplicación.

PO: ¿Y los taxistas cómo trabajan con esta aplicación?

ED: Como taxista también es necesario instalar una aplicación en su celular. El taxista debe estar registrado en una central que haya contratado el servicio de taxi-mobile. Cuando instale la aplicación se solicitarán sus datos identificatorios como taxista y de su móvil (nro. de taxi y dominio), y la selección de la central a la que pertenece. La aplicación también utiliza el sistema de posicionamiento que tiene el teléfono, para poder informar en qué lugar se encuentra el taxista.

PO: ¿Si estoy entendiendo bien, tanto el pasajero como el taxista deben contar con un Smart Phone con sistema de posicionamiento para que la aplicación funcione?

ED: Así es, sino la aplicación no puede ubicar al taxi ni al pasajero.

PO: ¿Y una vez que el pasajero sube al taxi, cómo se indica que ya está en viaje?

ED: El taxista debe asentar esto en la aplicación con la opción correspondiente. De esta forma la central está al tanto de la situación. Igualmente, cuando el pasajero descienda el taxista debe indicarlo, para que el taxi quede libre en la aplicación, y se muestre a un próximo pasajero.

PO: ¿La aplicación sólo muestra taxis libres?

ED: A los pasajeros sí, en la central se pueden ver con distintos colores los taxis ocupados, los libres y los que están fuera de servicio.

PO: ¿Y la aplicación para la central también es mobile?

ED: No, debería ser una aplicación web.

PO: ¿La idea es que la aplicación del pasajero y del taxista sean con touch screen?




ED: y los elementos centrales de la aplicación son el mapa y el taxi. Siempre a través de la selección del taxi se indicarán las acciones que quieren realizarse, como llamarlo, indicar que está ocupado, que está libre, tanto por parte del pasajero como del taxista.

A continuación, se muestran algunos prototipos de las interfaces que se espera vean los usuarios. La generación de los mapas será provista por la plataforma de Google Maps:



Solución Propuesta:

Se describen a continuación los principales roles de usuario:

Rol de Usuario	Descripción
 Pasajero	Persona que utilizará la aplicación con mucha frecuencia, para solicitar que le envíen un taxi al lugar donde está ubicado. Para él la simplicidad es importante. Está familiarizado con el uso de smartphones y puede instalar la aplicación. Su expectativa es que el taxi llegue a buscarlo lo más rápido posible.
 Taxista	Está familiarizado con el uso de smartphones y puede instalar la aplicación. Debe utilizar la aplicación mientras está trabajando, por eso necesita recibir solicitudes de viaje con notificaciones sonoras para enterarse de que tiene una solicitud de viaje. También debe poder utilizar la aplicación con manos libres. Necesita visualizar la posición del pasajero para poder llegar y necesita minimizar la cantidad de interacciones con el celular para informar que se dirige a buscar un pasajero o que ya lo encontró.
 Administrador de Central de Taxis	Debe poder trabajar con aplicaciones web y familiarizado con el uso de Google maps. Necesita poder visualizar los taxis con su estado y poder visualizar la localización de los pasajeros para poder asistir a los Taxistas. La información debe estar disponible en tiempo real.

<p>Loguear pasajero [user story canónica]</p> <p>Como Pasajero quiero loguearme para poder visualizar los taxis más cercanos.</p> <p>Notas:</p> <ul style="list-style-type: none"> • Datos para login: nombre, apellido, teléfono celular (opcional). • Pueden tomarse los datos de Facebook o desde el mismo celular. 	1
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar obtener datos desde Facebook con conexión a internet vía wifi. (pasa) <input type="checkbox"/> Probar obtener datos desde Facebook con conexión a internet vía 3g. (pasa) <input type="checkbox"/> Probar obtener datos desde Facebook sin conexión a internet. (falla) <input type="checkbox"/> Probar solicitar los datos al pasajero. (pasa) 	
<p>Loguear taxista</p> <p>Como Taxista quiero loguearme para poder visualizar los pedidos de taxis.</p> <p>Notas:</p> <ul style="list-style-type: none"> • Datos para login: nombre, apellido, teléfono celular, dominio, número de móvil, central. 	1
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar ingresar los datos completos cuando el taxista se encuentra asociado a una central. (pasa) <input type="checkbox"/> Probar ingresar los datos cuando el taxista no se encuentra asociado a una central. (falla) 	
<p>Registrar central de taxis</p> <p>Como Administrador de Central quiero dar de alta la central para poder tomar viajes con Taxi-mobile.</p>	1
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar registrar una central inexistente. (pasa) <input type="checkbox"/> Probar registrar una central existente. (falla) 	

<p>Pedir taxi</p> <p>Como Pasajero quiero poder pedir un taxi seleccionando el más conveniente de un mapa para asegurarme de que el taxi está cerca.</p> <p>Nota:</p> <ul style="list-style-type: none"> • El celular debe contar con el sistema de posicionamiento online. • El pasajero se debe visualizar gráficamente en un mapa. 	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar seleccionar el taxi de entre un conjunto de taxis visualizados (pasa) <input type="checkbox"/> Probar seleccionar un taxi cuando no existen taxis disponibles (realizando una selección en cualquier parte de la pantalla). (falla) <input type="checkbox"/> Probar visualizar los taxis cuando no está activo el sistema de posicionamiento del celular(falla) 	
<p>Ocupar taxi</p> <p>Como Taxista quiero marcar que el taxi se encuentra ocupado para no recibir pedidos de servicio que no podrá atender.</p>	2
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar ocupar el taxi cuando existe un viaje pedido pendiente. (pasa) <input type="checkbox"/> Probar ocupar el taxi cuando no existe un viaje pedido pendiente. (pasa) <input type="checkbox"/> Probar ocupar el taxi cuando ya está ocupado. (falla) 	
<p>Liberar taxi</p> <p>Como Taxista quiero liberar el taxi cuando estaba ocupado para que esté disponible para un próximo pedido de viaje.</p>	2
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar liberar el taxi cuando está con un viaje en curso. (pasa) <input type="checkbox"/> Probar liberar el taxi cuando está libre. (falla) 	

<p>Marcar taxi como fuera de servicio</p> <p>Como Taxista quiero marcar que el taxi se encuentra fuera de servicio para no recibir pedidos de servicio que no podrá atender.</p>	2
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar marcar el taxi como fuera de servicio cuando el taxi está libre. (pasa) <input type="checkbox"/> Probar marcar el taxi como fuera de servicio cuando existe un viaje pedido pendiente. (falla) <input type="checkbox"/> Probar marcar el taxi como fuera de servicio cuando ya está ocupado. (falla) 	
<p>Notificar a taxista y a central pedido de taxi</p> <p>Como Pasajero quiero enviar una notificación al momento en que solicito un viaje para que el taxista me busque y la central esté enterada del pedido.</p>	3
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar notificar al taxista y la central cuando ambos tienen conexión de internet. (pasa) <input type="checkbox"/> Probar notificar al taxista y la central cuando ninguno tiene conexión de internet. (falla) 	
<p>Buscar taxis cercanos</p> <p>Como Pasajero quiero ver cuáles con los 5 taxis más cercanos a mi ubicación para pedir el taxi que más me convenga.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Se muestra la ubicación del taxi y el tiempo estimado para llegar al lugar en el que se encuentra el pasajero. 	3
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar visualizar taxis cuando existen al menos 5 taxis libres. (pasa) <input type="checkbox"/> Probar visualizar taxis cuando no existe ningún taxi libre (falla) 	

<p>Ver taxi pedido</p> <p>Como Taxista quiero ver la ubicación del pasajero que ha solicitado un viaje para poder ir a buscarlo</p> <p>Nota:</p> <ul style="list-style-type: none"> • El celular debe contar con el sistema de posicionamiento online. • El pasajero se debe visualizar gráficamente en un mapa. 	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar visualizar un pedido de viaje pendiente. (pasa) <input type="checkbox"/> Probar visualizar un pedido de viaje cuando no se ha solicitado ninguno. (falla) <input type="checkbox"/> Probar visualizar un pedido de viaje cuando no está activo el sistema de ubicación del celular(falla) 	
<p>Ver mapa de taxis</p> <p>Como Administrador de la central de Taxis quiero ver la ubicación de todos los taxis de la central y si tienen viajes en curso para saber la disponibilidad actual</p>	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar visualizar un conjunto de taxis libres , fuera de servicio y ocupados (pasa) <input type="checkbox"/> Probar visualizar el mapa cuando ningún taxi de la central está conectado al sistema de ubicación (falla) 	

Nota: respecto a los puntos de historia se evalúa la consistencia en función de sus puntos de historia asignados, no se espera que coincidan con los indicados en la solución, son sólo a modo de guía para la corrección.

Tema: User Stories

Ejercicio 3: Caso Práctico – Sistema web mobile para el registro de gastos

Presentación del Caso de Estudio

Objetivo: Desarrollar un sistema para celulares y Tablet para que los usuarios puedan registrar sus gastos mensuales y mantener un control de los mismos.

A continuación, se transcribe parte de la entrevista realizada al experto en el dominio:

Product Owner (PO): ¿Cómo debe visualizarse la planilla de gastos?

Experto en el Dominio (ED): Similar a una tabla en Excel, con columnas donde se pueda indicar el monto, el tipo de gasto, y la fecha en que se realizó el gasto.

PO: ¿La fecha del gasto es la fecha actual? ¿Se toma automáticamente?

ED: Debería mostrarse la fecha actual, pero permitir modificarse.

PO: ¿Cuáles son tipos de gastos permitidos?

ED: Cada persona debería poder registrar sus tipos de gastos, así como indicar si el gasto es propio o de otra persona (por ejemplo, de su esposa).

PO: ¿El nombre o la relación con el usuario debe indicarse?

ED: Debe indicarse el nombre y apellido. También por defecto debe mostrarse el nombre y el apellido del usuario logueado, permitiendo modificarlo, y si alguna vez se registró el nombre y apellido que se comienza a ingresar (no importa si esta vez o una vez anterior), el sistema debería mostrar aquellos nombres y apellidos que comiencen de forma similar, como hace el Excel.

PO: Ah...¿Es decir que el usuario debe registrarse y cuando va a usar la aplicación debe iniciar sesión?¿Y la sesión caduca en algún momento?

ED: El usuario debe registrarse para permitirle descargar la aplicación, y la primera vez debe iniciar sesión pero luego se consideran los datos de la sesión registrados, salvo que el usuario decida desloguearse. La sesión no caduca nunca.

PO: ¿Y la planilla de gastos muestra todos los gastos? ¿Cómo se ordenan?

ED: Por defecto se deben mostrar todos los gastos del mes en curso ordenados desde el gasto más actual, pero el sistema debería permitir ver cualquier período que el usuario quiera, y que pueda filtrar por tipo de gasto, por responsable de gasto, por rango de montos. Además debe poder modificar el criterio de ordenamiento. Y para cada filtro que se aplique arriba se debe mostrar el total de gastos según el filtro aplicado.

Solución Propuesta

Roles de Usuario:

- ↔ Usuario Web Mobile
- ↔ Usuario

<p>Registrar gasto</p> <p>Como Usuario Web mobile quiero registrar un nuevo gasto para tener un seguimiento de mis gastos.</p> <p>Nota: Se debe poder indicar el monto, el tipo de gasto, y la fecha en que se realizó el gasto.</p>	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar registrar un nuevo gasto con todos los datos (pasa) <input type="checkbox"/> Probar registrar un nuevo gasto con algunos datos (pasa) <input type="checkbox"/> Probar registrar un nuevo gasto sin indicar monto (falla) 	
<p>Generar planilla de gastos</p> <p>Como Usuario Web mobile quiero visualizar mi planilla de gastos para conocer todos los gastos realizados en un período de tiempo.</p> <p>Nota: La visualización de la planilla debe ser similar a una tabla en Excel, con columnas donde se pueda indicar el monto, el tipo de gasto, y la fecha en que se realizó el gasto.</p>	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar generar una planilla con gastos registrados. (pasa) <input type="checkbox"/> Probar generar una planilla sin gastos registrados. (falla) 	
<p>Registrar usuario [userstory canónica]</p> <p>Como Usuario quiero darme de alta para poder comenzar a utilizar la planilla de gastos.</p>	1
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar registrar un usuario inexistente. (pasa) <input type="checkbox"/> Probar registrar un usuario existente. (falla) 	

Loguear usuario Como Usuario quiero loguearme para tener acceso a mi planilla de gastos.	1
Pruebas de Usuario <ul style="list-style-type: none"> <input type="checkbox"/> Probar loguear un usuario con un usuario y contraseña existente. (pasa) <input type="checkbox"/> Probar loguear un usuario con un usuario y contraseña inexistente. (falla) 	
Desloguear usuario Como Usuario quiero desloguearme para requerir mis datos de usuario una próxima vez que desee utilizar la planilla de gastos.	1
Pruebas de Usuario <ul style="list-style-type: none"> <input type="checkbox"/> Probar desloguear un usuario previamente logueado. (pasa) <input type="checkbox"/> Probar desloguear un usuario no logueado previamente. (falla) 	
Administrar tipos de gastos Como Usuario quiero registrar los diferentes tipos de gastos para agregar nuevos tipos cuando sea necesario.	2
Pruebas de Usuario <ul style="list-style-type: none"> <input type="checkbox"/> Probar registrar un tipo de gasto existente. (pasa) <input type="checkbox"/> Probar registrar un tipo de gasto inexistente. (falla) 	
Administrar responsables de gastos Como Usuario quiero registrar los responsables de gastos para agregar nuevos responsables cuando sea necesario.	2
Pruebas de Usuario <ul style="list-style-type: none"> <input type="checkbox"/> Probar registrar un responsable de gastos existente. (pasa) <input type="checkbox"/> Probar registrar un responsable de gastos inexistente. (falla) 	

<p>Filtrar y ordenar gastos (esta user story puede separarse en dos: Filtrar gastos y ordenar gastos)</p> <p>Como Usuario quiero filtrar mis gastos de la planilla de gastos y ordenarlos para facilitar la visualización.</p> <p>Nota:</p> <ul style="list-style-type: none">• Los filtros permitidos son: período, tipo de gasto, responsable de gasto, rango de montos.• Se puede ordenar por tipo de gasto, responsable, monto o fecha.• Para cada filtro que se aplique, arriba de la planilla se debe mostrar el total de gastos según el filtro aplicado.	<p>3</p>
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"><input type="checkbox"/> Probar filtrar los gastos por un tipo de gasto. (pasa)<input type="checkbox"/> Probar filtrar los gastos por un tipo de gasto, responsable, fecha y monto. (pasa)<input type="checkbox"/> Probar ordenar los gastos por responsable (pasa)<input type="checkbox"/> Probar ordenar los gastos por descripción. (falla)	

Tema: User Stories

Ejercicio 4: Caso Práctico – Sistema GPS

Presentación del Caso de Estudio

Objetivo: Desarrollar un sistema que permite a un conductor (entre otras funcionalidades), buscar un destino, obteniendo distintas alternativas para llegar hasta el punto marcado desde la ubicación actual.

A continuación, se transcribe parte de la entrevista realizada al experto en el dominio:

Product Owner (PO): ¿Cómo se puede buscar un destino deseado?

Experto en el Dominio (ED): La búsqueda puede realizarse en todos los mapas de las distintas ciudades, que serán cargados en el dispositivo, o bien en el mapa de una ciudad determinada.

PO: ¿Qué datos del destino son necesarios?

ED: Si desea buscar un destino por dirección, debería indicar primero el país y la ciudad, y luego ingresar el nombre de la calle y número.

PO: ¿Qué pasa si desconozco alguno de éstos datos, ¿Puedo buscar el destino por otros parámetros?

ED: Sí, también debe existir la posibilidad de buscar un destino mediante sus coordenadas, o indicando un cruce de calles. Las coordenadas se representan con tres números que indican longitud y tres números que indican latitud. Cada número representa los grados, minutos y segundos respectivamente. Además se debe indicar la orientación para la latitud (norte, sur) y para la longitud (este, oeste), de cada coordenada.

PO: ¿Me darías un ejemplo?

ED: Claro, por ejemplo 24° 45' 45'' Longitud Este – 45° 34' 23'' Latitud Sur.

PO: Gracias, está claro. ¿Mencionaste algo del cruce de calles?

ED: Sí, la búsqueda de un destino podría ser realizada por cruce de calles, primero debería ingresarse el nombre del país y de la ciudad de destino, y luego el nombre de las dos calles.

PO: Entendidos los parámetros de búsqueda, ¿Cómo se espera que se indique el camino?

ED: Se debe poder visualizar en el mapa el camino propuesto para dirigirse desde el punto actual (origen) hasta el destino señalado.

PO: Respecto del camino que debe visualizarse, ¿Alguna condición para las distintas alternativas?

ED: Ciertamente, antes de la visualización en el mapa, debería poder seleccionarse la ruta deseada: el camino más rápido, el camino más corto, el camino por caminos alternativos, el camino evitando peajes o el camino evitando controles. Además, una vez realizada la búsqueda, se debe permitir al conductor marcarla como favorita, ingresando si desea un nombre descriptivo, para que en caso de volver a necesitarla, evitar ingresar todos los datos nuevamente.

PO: ¿Para cualquier tipo de búsqueda?

ED: Así es, una vez encontrado el destino, se debe poder guardar el mismo en Favoritos, ya sea una dirección, un cruce de calles o coordenadas. Cuando el conductor desee dirigirse a un destino guardado con anterioridad, sólo debe consultar la opción Favoritos y buscar el destino deseado. Los destinos en Favoritos deberían visualizarse por orden alfabético según su nombre, y también deberían poder filtrarse deletreando el mismo. Para éste último se desea que a medida que el conductor ingrese el nombre, el

sistema vaya mostrando las opciones que contiene con los dígitos ingresados, como cuando se utiliza un buscador web.

PO: ¿Alguna otra información al respecto?

ED: Sí, que el sistema muestre al Conductor la velocidad promedio durante el viaje y la hora de llegada aproximada, actualizando ésta última en función de la velocidad. Y que te permita buscar un destino, a partir de las últimas búsquedas realizadas, por lo menos las últimas 5.

Solución Propuesta

Roles de Usuario:

↵ Conductor

<p>Generar Mapa con Camino</p> <p>Como Conductor quiero visualizar en el mapa el camino para llegar al destino deseado.</p> <p>Nota: El sistema deberá mostrar los diferentes caminos propuestos: recorrido más rápido, recorrido más corto, recorrido por caminos alternativos, recorrido evitando peajes (para destinos fuera de la ciudad origen) y recorrido evitando controles.</p>	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar mostrar el camino más corto (pasa) <input type="checkbox"/> Probar mostrar un camino alternativo (pasa) <input type="checkbox"/> Probar mostrar el camino evitando peajes (pasa) <input type="checkbox"/> Probar mostrar el camino más corto en un lugar sin conexión (falla) 	
<p>Buscar Destino por Dirección</p> <p>Como Conductor quiero buscar un destino a partir de una calle y altura para poder conocer las distintas alternativas de recorrido para llegar al destino deseado.</p> <p>Nota: la altura es el número de calle.</p>	3
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura existente (pasa). <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de una calle inexistente (falla). <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura inexistente (falla). <input type="checkbox"/> Probar buscar un destino en un país inexistente (falla). <input type="checkbox"/> Probar buscar un destino en País existente, ciudad inexistente (falla). 	

<p>Buscar Destino por Coordenadas</p> <p>Como Conductor quiero buscar un destino a partir de sus coordenadas para poder conocer las distintas alternativas de recorrido para llegar al destino deseado.</p> <p>Nota: Las coordenadas se representan con tres números que indican longitud y tres números que indican latitud. Cada número representa los grados, minutos y segundos respectivamente. Además, se debe indicar la orientación (norte, sur, este, oeste).</p>	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de dos coordenadas existentes (pasa). <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de una coordenada inexistente (falla). <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de dos coordenadas existentes sin indicar la orientación (falla). <input type="checkbox"/> Probar buscar un destino en un país inexistente (falla). <input type="checkbox"/> Probar buscar un destino en País existente, ciudad inexistente (falla). 	
<p>Buscar Destino por Cruce de Calles</p> <p>Como Conductor quiero buscar un destino mediante el cruce de dos calles para poder conocer las distintas alternativas de recorrido para llegar al destino deseado.</p>	3
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de dos calles que se cruzan (pasa). <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de dos calles paralelas (falla). <input type="checkbox"/> Probar buscar un destino en un país inexistente (falla). <input type="checkbox"/> Probar buscar un destino en País existente, ciudad inexistente (falla). <input type="checkbox"/> Probar buscar un destino en un país y ciudad existentes, de calles inexistentes (falla) 	
<p>Guardar Destino en Favoritos</p> <p>Como Conductor quiero guardar el destino buscado para poder utilizarlo nuevamente sin necesidad de ingresar los datos nuevamente.</p> <p>Nota: El sistema deberá permitir ingresar un nombre para identificar el destino a guardar.</p>	1
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar guardar un destino buscado ingresando un nombre descriptivo (pasa). <input type="checkbox"/> Probar guardar un destino buscado sin ingresar un nombre descriptivo (pasa). 	

<p>Buscar Destino en Favoritos</p> <p>Como Conductor quiero buscar un destino guardado para poder conocer el recorrido realizado para llegar al destino.</p>	2
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar buscar un destino guardado (pasa). <input type="checkbox"/> Probar buscar un destino no guardado (falla) 	

<p>Filtrar Destino en Favoritos</p> <p>Como Conductor quiero filtrar los destinos guardados por nombre para facilitar la búsqueda.</p> <p>Nota: A medida que el conductor ingrese el nombre, el sistema debe mostrar las opciones que contiene con los dígitos ingresados, como cuando se utiliza un buscador web</p>	2
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> <input type="checkbox"/> Probar filtrar los destinos guardados por nombre existente (pasa). <input type="checkbox"/> Probar filtrar los destinos guardados por ciudad (falla). 	

Tema: Revisiones Técnicas al Software

Ejercicio 1: Caso Práctico – Sistema GPS

Presentación de la porción de código

Nota: La clase no está completa, a finalidades del ejercicio tiene sólo 35 líneas.

```
1 //Clase Búsqueda
2 public class Busqueda
3 {
4     private date fechabusqueda;
5     private boolean bandera;
6     Private ArrayList<Destino> Destinos;
7     Public busqueda (Date f)
8     {
9         fechabusqueda = f;
10        Destinos = null;
11    }
12    private boolean AgregarDestinoComoFavorito (Destino Dest)
13    { //inserto el destino "Dest" en la posicion 1 en la lista de favoritos.
14        Boolean agregado = this.Destinos.add(1, Dest);
15        Destino.esFavorito = true;
16        System.out.println("Se agregó el Destino " + Dest.nombre + " como favorito.");
17        return true;
18    }
19 }
20 Public boolean quiarDestinoFavorito (Destino Dest)
21 {
22     String esFavorito = "";
23     Iterator i = Destinos.Iterator();
24     while (i.hasNext()){
25         Destino d = (Destino)i.next();
26         if(d == Dest)
27         {
28             Destinos.remove(d);
29             Dest.esFavorito = false;
30             Return true;
31         }
32         Else
33             return false;
34     }
35 }
36
```

Solución Propuesta:

Reporte de Inspección				
Fecha de Inicio:	16/11/2013			
Revisores:	Nombre	Rol		
	JM	Revisor/Moderador		
	CF	Lector		
	JR	Autor		
Alcance de la revisión:	Porción de código de clase Busqueda de la línea 1 a 33			
Productos de Trabajo Inspeccionados				
Nombre	Versión	Autor		
Clase Busqueda	Initial versión (ó 1.0)	JR		
Hallazgos				
Nº	Fecha	Descripción	Línea	Severidad
1	16/11/2013	La variable fechaBusqueda es declarada pero no utilizada	4	Menor
2	16/11/2013	La variable fechaBusqueda debería tener tipo de datos Date y debería importarse la librería correspondiente.	4	Mayor
3	16/11/2013	La variable bandera es declarada pero no utilizada	5	Menor
4	16/11/2013	La variable Destinos no respeta notación camelCase y el modificador de acceso debería escribirse en minúscula.	6	Menor*
5	16/11/2013	El nombre del constructor no es el mismo que el de la clase y el modificador de acceso debería escribirse en minúscula.	7	Mayor
6	16/11/2013	No se crea correctamente el arrayList de objetos Destinos. Debería crearse una instancia del arrayList: destinos= new ArrayList<>();	10	Mayor
7	16/11/2013	El modificador de acceso del método AgregarDestinoComoFavorito es incorrecto, debería ser public	12	Mayor
8	16/11/2013	El método AgregarDestinoComoFavorito no respeta notación camelCase, tanto en el nombre del método como en el nombre del parámetro.	12	Menor*
9	16/11/2013	El ArrayList Destinos es NULL por lo tanto no se pueden efectuar operaciones	14, 23, 24, 25 y 28	Mayor

10	16/11/2013	No se puede usar <tipo del objeto>.<atributo>	15	Mayor
11	16/11/2013	El atributo nombre del Destino está mal accedido (no se usa get)	16	Menor
12	16/11/2013	el metodo AgregarDestinoComoFavorito siempre devuelve el valor true	17	Mayor
13	16/11/2013	La variable esFavorito no es utilizada	22	Menor
14	16/11/2013	Falta cierre de llave del while del método quitarDestinoFavorito	24	Mayor
15	16/11/2013	La comparación de objetos es incorrecta, se debería comparar con un atributo o utilizando el método compareTo()	26	Mayor
16	16/11/2013	Mal accedida la variable esFavorito del objeto Dest (no utiliza método set)	29	Menor
17	16/11/2013	Incorrecta la sintaxis de la palabra return, no lleva mayúscula	30	Mayor
18	16/11/2013	Incorrecta sintaxis en return false, si no encuentra el primero ya corta el método	33	Mayor
19	16/11/2013	Faltan comentarios en general en toda la porción de código	NA	Cosmético
20	16/11/2013	Mal accedida la variable esFavorito del objeto Dest (no utiliza método set) y se pide el valor del atributo a la clase en lugar del objeto	15	Mayor
21	16/11/2013	Falta métodos de seteo en la clase	N/A	Menor
Decisión Final:		Aceptar/ Rechazar /Aceptar Provisoriamente		

* Se considera severidad menor (y no cosmético) por ser una violación al estándar especificado en el enunciado.

Ejercicio 2: Caso Práctico – Sistema de Afluencia

Presentación de la porción de código

Una empresa de servicios de la Ciudad de Córdoba ha decidido mudar sus oficinas comerciales y quiere aprovechar el cambio de locación para darle una nueva imagen a las mismas. Como parte de esta renovación se ha decidido desarrollar un sistema que permita gestionar la afluencia de los clientes, pudiendo además determinar la cantidad de clientes atendidos al final del día y el motivo del contacto.

Para este fin se conforma en el área de sistemas de la empresa un scrum team para que desarrolle el producto, quiénes comenzarán a trabajar en el mismo el primero de agosto.

El **PO** les explica que el sistema a construir deberá permitir a los clientes que se presenten en las oficinas de la empresa emitir un ticket de atención, mediante el cual serán llamados para ser atendidos. Este ticket

se imprimirá desde una terminal con pantalla táctil, en donde el cliente deberá seleccionar previamente el tipo de trámite a realizar y, en función de la opción seleccionada, el sistema imprimirá el ticket con una letra y un número.

La letra se asignará acuerdo al trámite a realizar, ya que en función de las mismas se administrarán distintas colas de atención. El número es el orden de atención en la cola de la letra. Por ejemplo: Para el tipo de trámite “Gestión Personalizada”, subtipo “Financiación” se asignará la letra F. Cada día se inicia el contador de atención en cero, con lo cual al primer cliente del día que requiere hacer un cambio de domicilio se le entrega un ticket con el orden F1, al segundo que se presente en el día se le entregará el ticket F2 y así sucesivamente.

Cada uno de los puestos de atención tendrá previamente asignada las letras que atiende, tarea que será realizada por el Supervisor del área (en función de las estadísticas de afluencia por tipo de trámite) y que puede ir cambiando en función de las necesidades del negocio. Por ejemplo, a los Puestos 1 y 2 el día 07/05/2014 se les asigna la letra F. Si hubiera clientes con un ticket con F sin atender, los Representantes de Atención (RA) ubicados en los Puestos 1 y 2 los irán llamando en función del orden de llegada (dados por el número impreso en el ticket)

En cada Puesto el RA indicará su estado: “Libre”, “Atendiendo”, “No disponible”. El estado “Atendiendo” se asignará automáticamente cuando desde el Puesto se llama al próximo cliente para ser atendido, el estado Libre se asume cuando se indica el fin de la atención y el estado No disponible es el que se utiliza cuando el RA no está en condiciones de atender clientes.

Cuando desde un Puesto de trabajo en estado Libre se llame al siguiente cliente, se informará por 2 pantallas LED el número a ser atendido y el Puesto al que debe dirigirse.

Además de organizar la atención de los clientes en el salón, el sistema deberá poder informar acerca del tiempo promedio de espera de tipo de trámite y el tiempo promedio de atención de los mismos, así como también tiempos máximos y mínimos de espera por rango horario y cantidad de clientes diarios atendidos por rango horario.


```

1 //Clase Ticket
2 import java.util.Date;
3 public class Ticket
4 {
5     private int numeroticket;
6     private TipoTramite tipoTramite;
7     private SubtipoTramite subtipoTramite;
8     private Date fechaEmision;
9     private char letra;
10    private Date tiempoEspera;
11
12    public Ticket()
13    {
14    }
15
16    public String conocerLetra(tipoTramite,SubtipoTramite)
17    {
18        //Falta resolver la asignación de la cola de atención según el tipo
19        //y subtipo de trámite
20        .....
21    }
22
23    public string conocerTipoTramite()
24    {
25        return TipoTramite.mostrarTipoTramite();
26    }
27
28    public String conocersubtipoTramite()
29    {
30        return subtipoTramite.mostrarTipoTramite();
31    }
32
33 }

```

```

35 //Clase TipoTramite
36
37 public class TipoTramite
38 {
39     private int ID;
40     private String descripcion;
41
42     public TipoTramite()
43     {
44     .....
45     }
46     public String conocerTipoTramite()
47     {
48         return System.out.println(descripcion);
49     }

```

```

50
51 //Clase subtipoTramite
52 public class subtipoTramite
53 {
54     private String descripcion;
55
56     public subtipoTramite()
57     {
58     }
59
60     public String conocerSubtipoTramite()
61     {
62         //public String getDescripcion()
63     }
64 }

```

Propuesta de solución

Reporte de Inspección				
Fecha de Inicio:				
Revisores:	Nombre	Rol		
	GP, LC	Autores		
	Juan, Manuel	Inspector		
	Manuel	Lector		
	Carlos	Moderador		
	José	Anotador		
Alcance de la revisión:	Clases del paquete "Emitir Ticket"			
Nombre	Versión	Autor		
Clase Ticket	2.0	GP		
Clase TipoTramite	1.0	LC		
Clase SubtipoTramite	1.0	LC		
Hallazgos				
Nº	Fecha	Descripción	Línea	Severidad
1	17/06/2014	El atributo numeroticket no se utiliza	5	Menor
2	17/06/2014	El atributo numeroticket no respeta la notación camelCase	5	Cosmético
3	17/06/2014	El atributo fechaEmision no se utiliza	8	Menor
4	17/06/2014	El atributo letra no se utiliza	9	Menor
5	17/06/2014	El atributo tiempoEspera no se utiliza	10	Menor
6	17/06/2014	El método conocerLetra está incompleto, tiene un comentario que indica que aún falta resolver asignación de la cola de atención. Además la signatura está mal formada: los parámetros llevan tipo de dato y nombre.	18	Mayor
7	17/06/2014	El método conocerLetra no está retornando ningún valor cuando su signatura indica que debería devolver un String.	16	Mayor
8	17/06/2014	Está mal declarado el retorno del método	23	Mayor
9	17/06/2014	El atributo ID no respeta la notación camelCase	39	Cosmético
10	17/06/2014	Hay una llave de apertura de un método sin llave de cierre	43	Mayor
11	17/06/2014	El método conocerTipoTramite no está retornando ningún valor	45	Mayor
12	17/06/2014	El nombre de la clase subTipoTramite no respeta la notación camelCase	52	Cosmético
13	17/06/2014	No debería existir código comentado	62	Menor
14	17/06/2014	No debería existir un método sin implementación	62	Mayor

15	17/06/2014	Las variables no están inicializadas en la clase Ticket	5-10	Menor
Decisión Final:		Aceptar/ Rechazar /Aceptar Provisoriamente		

Ejercicio 3: Caso Práctico – Empresa de Televisión por Cable

El área de desarrollo de sistemas de una empresa de televisión por cable se encarga de realizar desarrollos para los sistemas que dan soporte a la prestación de servicios de la empresa. En este momento se les ha presentado el desafío de realizar un desarrollo innovador, para brindar a los usuarios del servicio de cable una interfaz de consulta de programación televisiva ágil y con diferentes formas de búsqueda, entre las que se incluye: búsqueda por tipo de programación (películas, series, informativos, deportes, etc), búsqueda por nombre de la programación, búsqueda por elenco, búsqueda por tipo de visualización (HD o Analógica), etc. Al tratarse de un proyecto de innovación, ésta área ha definido trabajar con un framework scrum, donde el Gerente de Desarrollo fue designado como el Product Owner, debido a que cuenta con mucha experiencia en la empresa, conoce los servicios y el funcionamiento actual de los sistemas. A esto se le suma que la idea original del proyecto fue suya.

El equipo de proyecto trabaja de la siguiente manera: Se realizan reuniones con el Product Owner para definir los requerimientos, los cuales se registran en un sistema como tarjetas que representan user stories. Estas user stories forman parte del product backlog. Para definir la arquitectura de la solución propuesta se realizan draft en un pizarrón con el que cuenta el equipo, y se le sacan fotos para guardar las propuestas realizadas. Dichas propuestas de arquitectura se guardan en el repositorio del producto, siendo versionadas cada vez que surge alguna modificación. Cabe aclarar que el área cuenta con la herramienta Clear Case, por lo que el versionado de todo el código fuente y documentación necesaria lo maneja la propia herramienta. Se está trabajando con la modalidad de vistas dinámicas, por lo que siempre se cuenta con una vista actualizada a toda la información. En ocasiones la empresa ha definido que para realizar un control de los ítems de configuración, se registra al final de cada iteración de un proyecto la cantidad de ítems de configuración con los que se cuenta hasta este momento.

Para generar el código se cuenta con una herramienta con interfaz gráfica de desarrollo sincronizada con Clear Case, de tal forma de ir manteniendo el versionado del código fuente en el repositorio del producto. A su vez, la herramienta cuenta con opciones para compartir el escritorio, lo cual facilita la ejecución de revisiones de pares del código generado. Estas revisiones se ejecutan y los hallazgos identificados por el revisor son resguardados para luego realizar un seguimiento y verificar que se corrijan dichos hallazgos. Puede requerirse contabilizar la cantidad de hallazgos por severidad, para determinar si la calidad en el producto mejora luego de realizar las correcciones definidas en cada revisión.

Cuando se realizan las meeting planning, en el pizarrón que se utiliza para gestionar el proyecto se colocan postit que identifican cada actividad a realizar, el responsable y la estimación correspondiente a la tarea. Una vez armado el pizarrón se le saca una fotografía para contar con la información correspondiente al inicio de un sprint. Además se importan en pdf desde la herramienta las user stories asociadas a las tareas a realizar, para contar con los requerimientos que se trabajarán en el sprint. En estas meeting planning se define la capacidad del equipo, para poder estimar el trabajo a realizar.

Cuando se comienza a ejecutar el proyecto, al realizar las daily meeting en un pizarrón se genera el burn down chart para ir conociendo el trabajo ya “quemado”. Finalmente en las sprint review se compara el gráfico burn down chart con la capacidad definida para el sprint.

Presentación de la porción de código

Nota: La clase no está completa, a finalidades del ejercicio tiene sólo 20 líneas.

```
1 class Busq{
2     //@autor CM
3     //private List <CriterioBusqueda> criteriosBusqueda;
4     private boolean hd;
5     public List <TipoProgramacion> tipos;
6
7     //@Method Constructor por defecto
8     public Busq() {
9         // criteriosBusq = new List();
10        hd = false;
11        tipos = null; }
12
13    private TipoProgramacion busquedaportipoprogramacionyVisualizacion(boolean
14    visualizacion; Tipoprogramación tipo) {
15        Iterator i = tipos.Iterator();
16        while (i.hasNext()){ TipoProgramacion tipoActual = (TipoProgramacion)i.next();
17        if ( hd && tipoActual == tipo) return tipo;
18        else { if (!hd&& tipos.getValue(i)==tipo) return tipo;}}
19    return null;}
20 }
```

Solución Propuesta

Reporte de Inspección				
Fecha de Inicio:	09/11/2012			
Revisores:	Nombre	Rol		
	XXX YYY	Revisor		
	CM	Autor		
Alcance de la revisión:	19 líneas de la Clase Busq			
Productos de Trabajo Inspeccionados				
Nombre	Versión	Autor		
Clase Busq	1.5	CM		
Hallazgos				
Nº	Fecha	Descripción	Línea	Severidad
1	17/11/2012	El nombre de la clase no es claro.	1	Cosmético
2	17/11/2012	No debería existir código comentado	3	Leve
3	17/11/2012	No es claro el nombre de la variable hd	4	Cosmético
4	17/11/2012	No es claro el nombre de la variable tipos	5	Cosmético
5	17/11/2012	Para poder declarar una List se debe importar la librería: java.util.List.	5	Mayor
6	17/11/2012	Inicialización de la lista incorrecta. List es la interfaz, se debería crear una instancia de ArrayList por ejemplo: tipos = new ArrayList<>();	11	Mayor
7	17/11/2012	El modificador de acceso del método busquedaortipopogramacionyVisualizacion no debería ser privado ya que no permite el acceso al método desde otras clases.	13	Mayor
8	17/11/2012	El nombre del método busquedaortipopogramacionyVisualizacion no cumple la notación camel case	13	Cosmético
9	17/11/2012	Se intenta acceder a un método de un objeto nulo (la lista de tipos), al realizar la operación tipos.iterator()	15	Mayor
10	17/11/2012	Las sentencias if and else realizan la misma acción, por lo que no realiza ningún aporte contar con dichas sentencias.	17	Mayor
11	17/11/2012	Se intenta acceder a un método del iterator: i.hasNext() de una lista que está valiendo null (la lista de tipos).	16	Mayor
12	17/11/2012	No debería existir código comentado	9	Leve
Decisión Final:		Aceptar/ Rechazar /Aceptar Provisoriamente		

Ejercicio 4: Caso Práctico – Desarrollos a Medida

Presentación de la porción de código

Nota: La clase no está completa, a finalidades del ejercicio tiene sólo 32 líneas.

```
1  import java.util.ArrayList;
2  //Clase Localidad
3  public class Localidad {
4      private String DESCRIPCION;
5      private String nombre;
6      private ArrayList<Barrio> barrios;
7
8      public Localidad(String nomb) {
9          nombre = nomb;
10     }
11
12     public String getNombre() {
13         return nombre;
14     }
15     public void setNombre(String nombre) {
16         this.nombre = nombre;
17     }
18
19     public Barrio[] getBarrios() {
20         return barrios;
21     }
22
23     public void addBarrio(Barrio barrio) {
24         int banderaA = 0;
25         if(this.barrios == null){
26             this.barrios = new Barrio();
27         }
28         boolean banderaB = this.barrios.add(barrio);
29
30         return bandera;
31     }
32 }
```

Solución Propuesta:

Reporte de Inspección			
Fecha de Inicio:	09/11/2012		
Revisores:	Nombre	Rol	
	XXX YYY	Revisor	
	CM	Autor	
Alcance de la revisión:	19 líneas de la Clase Busq		
Productos de Trabajo Inspeccionados			
Nombre	Versión	Autor	
Clase Busq	1.5	CM	
Hallazgos			
Nº	Descripción	Referencia Ubicación	Severidad
1	La variable DESCRIPCIÓN no respecta la notación camelCase	Línea 4	Baja
2	La variable DESCRIPCION no se inicializa ni se utiliza	Línea 4	Media
3	El arraylist barrios no se inicializa	Línea 8	Alta
4	El método addBarrio() está declarado con retorno void, pero está retornando un valor.	Línea 30	Alta
5	La variable "bandera" que retorna el método no fue usada en toda la operación	Línea 30	Alta
6	La variable barrios es un ArrayList y el método getBarrios está retornando un vector. Tipos de datos incompatibles	Línea 19	Alta
Decisión Final:	Aceptar/ Rechazar /Aceptar Provisoriamente		

Tema: Administración de Configuración de Software

Ejercicio 1: Caso Práctico – GPS

Presentación del Caso de Estudio

Consideraciones para el Proyecto

Para construir el software, la empresa desarrolladora utilizará un proceso de desarrollo ágil, específicamente con un framework SCRUM. El equipo de proyecto trabaja de la siguiente manera: Cuando la información no es clara o está incompleta, como en este caso, el Product Owner mantiene entrevistas con los expertos del dominio, para luego realizar reuniones con el equipo del proyecto para definir los requerimientos, los cuales se registran en un sistema como tarjetas que representan user stories.

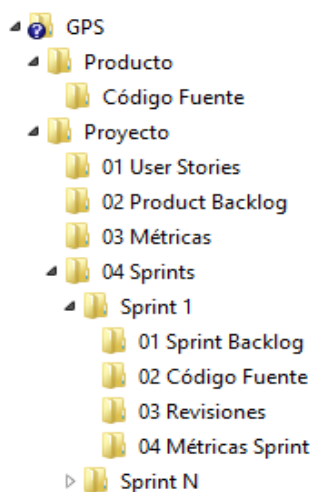
Cuando se realizan las reuniones de planificación (planning meeting), en el pizarrón se colocan post-its que identifican cada actividad a realizar, el responsable y la estimación correspondiente a la tarea. Una vez armado el pizarrón se le saca una fotografía para contar con la información correspondiente al inicio de un sprint.

Durante la ejecución del sprint al realizar las daily meeting en un pizarrón se genera el burndown chart para ir conociendo el avance del trabajo y se genera, si es necesario la lista de impedimentos (blocks list) y se gestiona la lista de riesgos identificados. Luego en las sprint review meetings se compara el gráfico burndown chart con la capacidad definida para el sprint.

Cabe aclarar que el área cuenta con la herramienta Clear Case, por lo que el versionado de todo el código fuente y documentación necesaria se realiza con esta herramienta. Se está trabajando con la modalidad de vistas dinámicas, por lo que siempre se cuenta con una vista actualizada de toda la información.

Para generar el código se cuenta con una herramienta con interfaz gráfica de desarrollo sincronizada con Clear Case, de tal forma de ir manteniendo el versionado del código fuente en el repositorio del producto. A su vez, la herramienta cuenta con opciones para compartir el escritorio, lo cual facilita la ejecución de revisiones de pares del código generado. Estas revisiones se ejecutan y los hallazgos identificados por el revisor son resguardados para luego realizar un seguimiento y verificar que se corrijan dichos hallazgos.

Propuesta de Solución

Estructura del Repositorio**Ítems de Configuración:**

Nombre del Ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de Ítem [Producto / Proyecto / Iteración]
Sprint Backlog (con sus user stories)	GPS_Sprint_N_Backlog	GPS/Proyecto /01 User Stories	Producto
Revisión de Pares	Revisión_Par_aaaammdd	GPS/Proyecto /04 Sprints /Sprint 1 /03 Revisiones	Iteración
ProductBacklog	GPS_Product_Backlog	Herramienta de gestión de backlog	Producto
Código fuente	NombreClase	GPS/Producto /Código Fuente	Producto
Métricas del Proyecto	GPS_Métrica_NombreMétrica	GPS/Proyecto /03 Métricas	Proyecto

Ejercicio 2: Caso Práctico – Herramienta de Seguimiento de Defectos**Presentación del Caso de Estudio**

Una empresa se dedica al desarrollo y mantenimiento de software a medida para distintos clientes. Los productos de software que se construyen son basados en Windows. El motivo por el cual se requiere el producto, es porque luego de haber implementado varios sistemas para distintos clientes, han detectado que los productos de software que entregan presentan defectos que no son encontrados en las etapas de desarrollo.

Esto hace que sus clientes tengan que registrar los errores en papel, enviar un mail, o bien, hacer un llamado telefónico para informar los detalles del defecto encontrado al equipo de desarrollo.

Haciendo un análisis hacia adentro, también han detectado que los equipos de desarrollo de la empresa no cuentan con un proceso de registro, evaluación, asignación y seguimiento de los defectos que se detectan en las etapas de desarrollo.

Es por esto que han decidido implementar un sistema de seguimiento de defectos (*Bug Tracking System*) sencillo que se adapte a sus necesidades puntuales.

Con este sistema se pretende:

Guía Práctica del Módulo Desarrollo de Software

Versión 1.0 – Liberada el 15/02/2017

- Establecer un proceso e implementarlo a través de una herramienta que permita gestionar los defectos encontrados de manera de tratar de reducir la cantidad de defectos que son detectados por los usuarios finales
- En el caso de que el usuario final detecte un defecto, contar con un mecanismo simple y uniforme para que el usuario pueda informar la existencia de los mismos.

El mismo sistema debería permitir registrar los defectos de todas las aplicaciones que están en desarrollo, aplicaciones instaladas en los distintos clientes y nuevas aplicaciones a desarrollar.

Desde ya que cada usuario (cliente o desarrollador) solo debe poder operar con la información de defectos correspondientes a las aplicaciones con las que está asociado.

Debido a que no solo los desarrolladores sino también los clientes de la empresa deben poder acceder al sistema, se necesita que el mismo esté disponible a través de Internet. Dicha disponibilidad debería ser permanente durante el horario de trabajo de la empresa, como así también durante el horario de trabajo de sus clientes.

Otro punto importante a considerar es que el manejo del sistema debe ser simple, para que cualquier persona con conocimientos básicos de manejo de PC lo pueda utilizar; y rápido, principalmente en el momento de registrar los defectos.

Con la intención de obtener a corto plazo una versión funcional del producto, que luego se irá extendiendo y mejorando a partir de la validación con sus clientes, se ha decidido emplear un ciclo de vida Iterativo e Incremental para este proyecto, definiendo tres iteraciones para completar el producto.

El proyecto tiene previsto realizar entrevistas para relevar los requerimientos de los distintos clientes; las que podrán repetirse en cada iteración según necesidad, generando la actualización de la Especificación de Requerimientos de Software del producto (ERS). Todas las entrevistas deberán estar documentadas en minutas de relevamiento.

El proyecto incluye despliegue del sistema, su parametrización y puesta a punto para comenzar a trabajar, que incluye la carga inicial y la parametrización. La carga de datos inicial es mucha y para la época en que se realicen esas actividades está previsto que el personal se tome vacaciones, ya que sería a fines de diciembre o primeros días de enero. Las personas relevadas hasta el momento han puesto muchas expectativas sobre los beneficios del sistema y son todos muy sensibles al aspecto estético que tengan las pantallas y los reportes, ya que casi todos los empleados no son especialistas en el manejo de software.

Se está evaluando utilizar como lenguaje de programación VB.Net, pero se requerirá una capacitación especial para los programadores y algunos otros miembros del equipo. El presupuesto inicial que se pasó por el desarrollo no tenía prevista la contratación de terceros, por lo que el Líder de Proyecto deberá estar muy atento para no exceder su presupuesto. Se empleará la herramienta Subversion para el control de versiones de los ítems de configuración y líneas base del proyecto.

Es necesario dimensionar los recursos necesarios para las pruebas, ya que no se sabe si el hardware disponible para pruebas será el necesario y suficiente para probar el producto en la etapa de prueba de sistemas.

Como resultado de la Planificación, se generarán los siguientes documentos: Estimación, WBS, Cronograma del Proyecto (Gantt), Plan de Desarrollo de Software, Plan de Iteración y Planes de Soporte (Plan de Gestión de Riesgos, Plan de Calidad, Plan de Prueba del Proyecto, Plan de Despliegue, Plan de Aceptación del Producto, Plan de Gestión de Configuración de Software, Plan de Participación de Involucrados).

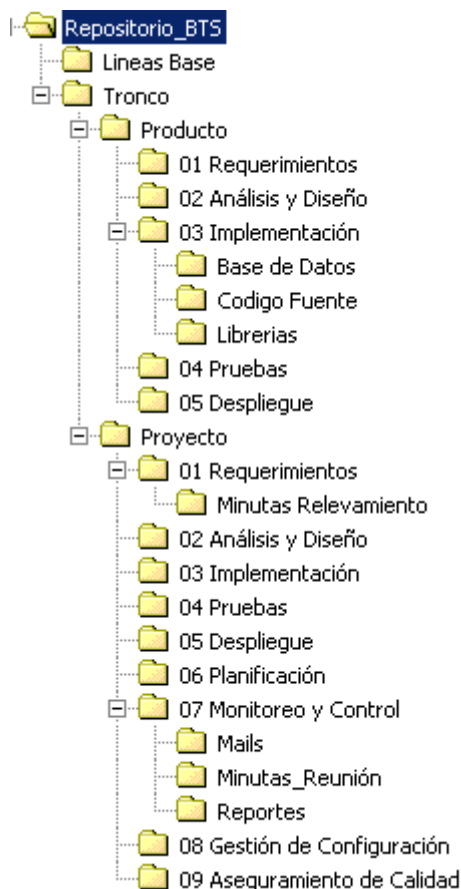
La Gerencia ha solicitado que cada 15 días el Líder de Proyecto informe sobre el avance del proyecto, para lo cual deberá generar un Reporte de Control de Estado.

Las reuniones de seguimiento del proyecto deberán estar documentadas bajo las respectivas minutas de reunión, dejando constancia del tema tratado, decisiones tomadas, fecha y hora de inicio, participantes y duración de la misma.

El plazo máximo estipulado para que el producto esté funcionando y en régimen es de 6 meses.

Propuesta de Solución

Estructura de Repositorio



Listado de Ítems de Configuración

Nombre del Ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de Ítem [Producto / Proyecto / Iteración]
Especificación de Requerimientos de Software	BTS_ERS	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/01 Requerimientos">http://<nro servidor>/BTS/Tronco/Proyecto/01 Requerimientos	Producto
Minuta Relevamiento	BTS_MINUTA_RELEVAMIENTO_<ITERxx>_<YYYYMMDD>_<HHMM>	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/01 Requerimientos/Minutas Relevamiento">http://<nro servidor>/BTS/Tronco/Proyecto/01 Requerimientos/Minutas Relevamiento	Iteración
Checklist/Informe de Revisión de QA de Requerimientos	BTS_CHK_REV_QA_REQ_<NN>_<YY YYMMDD>	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/01 Requerimientos">http://<nro servidor>/BTS/Tronco/Proyecto/01 Requerimientos	Proyecto
Matriz de Rastreabilidad	BTS_M_RASTREABILIDAD	<a href="http://<nro servidor>/BTS/Tronco/Producto/01 Requerimientos">http://<nro servidor>/BTS/Tronco/Producto/01 Requerimientos	Producto

Nombre del Ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de Ítem [Producto / Proyecto / Iteración]
Casos de Uso	BTS_CU_<NOMBRE CU>_<Nro. CU>	http://<nro IP servidor>/BTS/Producto/Tronco/01 Requerimientos	Producto
Diagrama de Casos de Uso	BTS_DIAG_CU	http://<nro IP servidor>/BTS/Tronco/Producto/01 Requerimientos	Producto
Casos de Prueba	BTS_C_PRUEBA	http://<nro IP servidor>/BTS/Tronco/Producto/04 Pruebas	Producto
Documento de Diseño	BTS_DOC_DISEÑO	http://<nro IP servidor>/BTS/Tronco/Producto/02 Análisis y Diseño	Producto
Componente de Software (código fuente, ejecutables)	BTS_COMPONENTE_SW	http://<nro IP servidor>/BTS/Tronco/Producto/03 Implementación/Código Fuente	Producto
Librerías comunes	BTS_<LIBRERÍA>	http://<nro IP servidor>/BTS/Tronco/Producto/03 Implementación/Librerías	Producto
Componente de Datos (base de datos, scripts)	BTS_<COMPONENTE_DATOS>	http://<nro IP servidor>/BTS/Tronco/Producto/03 Implementación/Base de Datos	Producto
Nota de Release	BTS_NOTA_RELEASE_<ITERxx>	http://<nro IP servidor>/BTS/Tronco/Proyecto/05 Despliegue	Iteración
Plan de Despliegue	BTS_PLAN_DESPLIEGUE	http://<nro IP servidor>/BTS/Tronco/Proyecto/05 Despliegue	Proyecto
Plan de Aceptación de Producto	BTS_PLAN_ACEPT_PRODUCTO	http://<nro IP servidor>/BTS/Tronco/Proyecto/05 Despliegue	Proyecto
Planilla de Estimación	BTS_P_ESTIMACION	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto
Plan de Riesgos	BTS_PLAN_RIESGOS	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto
Checklist de Identificación de Riesgos	BTS_CHK_IDEN_RIESGOS	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto
Plan de Desarrollo de Software	BTS_PLAN_DESARROLLO_SW	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto
Plan de Iteración	BTS_PLAN_ITERACION_<ITERxx>	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Iteración
Cronograma del Proyecto	BTS_C_GENERAL	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto
Plan de Pruebas	BTS_PLAN_PRUEBAS	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto

Nombre del ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de ítem [Producto / Proyecto / Iteración]
Checklist/Informe de Revisión de QA de la Planificación	BTS_CHK_REV_QA_PLANIF_<NN>_<YYYYMMDD>	http://<nro IP servidor>/<Nombre Proyecto>/Tronco/Proyecto/06 Planificación	Proyecto
Aprobación de Requerimientos por Cliente	BTS_AP_REQ_CLIENTE_<YYYYMMDD>_<HHMM>	http://<nro IP servidor>/<Nombre del Proyecto>/Tronco/Proyecto/01 Requerimientos	Proyecto
Aprobación de Requerimientos por Gerencia	BTS_AP_REQ_GEREN_<YYYYMMDD>_<HHMM>	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto
Aprobación de Planes por Cliente	BTS_MAIL_AP_PLANES_CLIENTE_<YYYYMMDD>_<HHMM>	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto
Aprobación de Planes por Gerencia	BTS_MAIL_AP_PLANES_GEREN_<YYYYMMDD>_<HHMM>	http://<nro IP servidor>/BTS/Tronco/Proyecto/06 Planificación	Proyecto
Mails del Proyecto	BTS_MAIL_<ASUNTO>_<YYYYMMDD>	http://<nro IP servidor>/BTS/Tronco/Proyecto/07 Monitoreo y Control/Mails	Proyecto
Minutas de Reunión del Proyecto	BTS_MINUTA_<ASUNTO>_<YYYYMMDD>_<HHMM>	http://<nro IP servidor>/BTS/Tronco/Proyecto/07 Monitoreo y Control/Minutas_Reunión	Proyecto
Planillas de Recolección y Análisis de Métricas	BTS_P_METRICAS_<ITERxx>	http://<nro IP servidor>/BTS/Tronco/Proyecto/09 Aseguramiento de Calidad	Iteración
Checklist/Informe de la Auditoría del Proceso	BTS_CHK_INF_AUD_PROCESO_<ITERxx>_<NN>_<YYYYMMDD>	http://<nro IP servidor>/BTS/Tronco/Proyecto/07 Monitoreo y Control	Iteración
Plan de Calidad	BTS_PLAN_QA	http://<nro IP servidor>/BTS/Tronco/Proyecto/09 Aseguramiento de Calidad	Proyecto
Reporte de Estado de Ítems de Configuración	BTS_REP_EST_ITEMS_CONF_<NN>_<YYYYMMDD>	http://<nro IP servidor>/BTS/Tronco/Proyecto/08 Gestión de Configuración	Proyecto
Línea Base del proyecto	BTS_L_BASE_<ITERxx>_<NOMBRE_LB>_<NN>	http://<nro IP servidor>/BTS/ Líneas Base	Iteración
Documento de Línea Base	BTS_DOC_L_BASE_<ITERxx>_<NOMBRE_LB>_<NN>	http://<nro IP servidor>/BTS/ Líneas Base/BTS_L_BASE_<ITERxx>_<NOMBRE_LB>	Iteración
Plan de Administración de Configuración	BTS_PLAN_ADM_CONF	http://<nro IP servidor>/BTS/Tronco/Proyecto/08 Gestión de Configuración	Proyecto
Checklist/Informe de la Auditoría de la Configuración Física	BTS_CHK_AUD_CONF_FIS_<ITERxx>_<NN>_<YYYYMMDD>	http://<nro IP servidor>/BTS/Tronco/Proyecto/08 Gestión de Configuración	Proyecto

Nombre del ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de ítem [Producto / Proyecto / Iteración]
Checklist/Informe de la Auditoría de la Configuración Funcional	BTS_CHK_AUD_CONF_FUN_<ITERxx>_<NN>_<YYYYMMDD>	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/08">http://<nro servidor>/BTS/Tronco/Proyecto/08 Gestión de Configuración	IP Iteración
Checklist Cierre de Proyecto/Iteración	BTS_CHK_CIERRE_PROY_ITER_<ITERxx>_<YYYYMMDD>	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/07">http://<nro servidor>/BTS/Tronco/Proyecto/07 Monitoreo y Control	IP Iteración
Reporte de Control de Estado para Gerencia	BTS_REPORTE_EST_GCIA_<ITERxx>_<YYYYMMDD>	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/07">http://<nro servidor>/BTS/Tronco/Proyecto/07 Monitoreo y Control/Reportes	IP Iteración
Reporte de Control de Estado para Cliente	BTS_REPORTE_EST_CLIENTE_<ITERxx>_<YYYYMMDD>	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/07">http://<nro servidor>/BTS/Tronco/Proyecto/07 Monitoreo y Control/Reportes	IP Iteración
Reporte de Post Mortem	BTS_REP_POST_MORTEM_<ITERxx>	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/07">http://<nro servidor>/BTS/Tronco/Proyecto/07 Monitoreo y Control/Reportes	IP Iteración
Plan de Medición	BTS_PLAN_MEDICION	<a href="http://<nro servidor>/BTS/Tronco/Proyecto/07">http://<nro servidor>/BTS/Tronco/Proyecto/07 Monitoreo y Control	IP Proyecto

Sigla	Significado
<nro IP servidor>	Número de IP del servidor en el que se encuentran el repositorio del Proyecto BTS. Ej: 172.150.1.33
<ITERxx>	Número de la iteración del proyecto. Ej: ITER02
<ASUNTO>	Asunto del Mail o Minuta
<NN>	Número cardinal comenzando en 00.
<YYYYMMDD>	Fecha en formato numérico (AñoMesDía).
<HHMM>	Hora de inicio en formato numérico (HoraMinutos)
<NOMBRE_LB>	Nombre de la Línea Base. Se indica generalmente el Tipo de Línea Base. Ej: Fin de Implementación

Ejercicio 3: Caso Práctico – Liga Regional de Fútbol

Presentación del Caso de Estudio

La Asociación Cordobesa de Fútbol ha contratado a una empresa que se dedica al desarrollo de software a medida para distintos clientes, con el objetivo de contar con un producto de software que le ayude con la administración de los campeonatos (y los clubes que participan en los mismos) que organiza a través de las Ligas Regionales de Fútbol.

Todos los años se realizan campeonatos de fútbol. Los clubes que desean participar pueden inscribirse en la Liga hasta un mes antes del comienzo del campeonato.

Luego de la inscripción cada club debe presentar la lista de sus jugadores, un examen médico para cada uno, y los datos de la cancha habilitada para los partidos en los que participe como local. Cuando se presenta dicha lista el jugador queda registrado para participar con ese club en el campeonato. No obstante, hasta que no presente el examen médico no estará habilitado para jugar.

Cuando la Liga realiza la diagramación de un campeonato, de acuerdo al número de clubes inscriptos, determina la cantidad de jornadas o “fechas” a realizarse, y para cada una se definen los partidos a jugarse. Además, para cada partido se indica la cancha en que se realizará, los árbitros que intervendrán y el rol de cada uno en ese partido (árbitro principal, juez de línea, etc.).

Luego de culminada cada fecha, deben registrarse los resultados de cada partido:

- Goles convertidos: Indicando para cada gol el jugador que lo realizó y el tiempo de juego transcurrido.
- Amonestaciones efectuadas: Indicando para cada una el tipo de tarjeta (roja o amarilla), el jugador y el tiempo de juego transcurrido.

La Asociación Cordobesa de Fútbol desea tener un registro para cada jugador del tiempo jugado en cada partido y asignarle una calificación en cada partido, para luego poder premiar o reconocer de alguna manera a aquellos con el promedio más alto y la mayor cantidad de minutos jugados en un campeonato.

Además, se desea emitir una lista de goleadores (jugadores que convirtieron más goles) para premiar al goleador de cada campeonato. Para todos los reportes y estadísticas se ha requerido que los mismos se muestren tanto en formato de tablas como en formato gráfico. Todos los involucrados han puesto muchas expectativas sobre los beneficios del sistema y son todos muy sensibles al aspecto estético que tengan las pantallas y los reportes, ya que la mayoría de los empleados no son especialistas en el manejo de software.

Para realizar el desarrollo del sistema, la empresa desarrolladora utilizará un proceso de desarrollo tradicional. Con la intención de obtener a corto plazo una versión funcional del producto, que luego se irá extendiendo y mejorando, se ha decidido emplear un ciclo de vida Iterativo e Incremental para este proyecto, definiendo tres iteraciones para completar el producto.

El alcance del proyecto implica la especificación de los requerimientos, el desarrollo del producto, las pruebas, el despliegue del mismo y la capacitación a un representante de cada una de las Ligas Regionales. Es fundamental para el éxito del proyecto la aceptación del producto no sólo por parte de los referentes de cada Liga Regional sino también por los usuarios de cada uno de los clubes, dado que se ha decidido que el software permita que cada club pueda gestionar un usuario y una clave de acceso y pueda registrar su propia inscripción y la inscripción de la lista de jugadores que participará. Como consecuencia, el producto de software no sólo deberá desarrollarse con tecnología web, sino que el manejo del sistema debe ser simple para que cualquier persona con conocimientos básicos de manejo de PC lo pueda utilizar; y rápido, principalmente en el momento de registrar los datos de la inscripción.

El plazo expuesto para la finalización del proyecto es de 40 semanas y se requiere que además se contemple un período de “garantía” de 15 días hábiles. En el contrato están estipuladas multas que la empresa deberá pagar por incumplimiento de plazos. Además, el presupuesto inicial que se pasó por el desarrollo no tenía prevista la contratación de terceros. Por lo que el Líder de Proyecto deberá estar muy atento para no excederse en tiempos y en el presupuesto.

Se está evaluando utilizar como lenguaje de programación VB.Net, pero se requerirá una capacitación especial para los programadores y algunos otros miembros del equipo. Se emplearán las herramientas Subversion y TortoiseSVN para el control de versiones de los ítems de configuración y líneas base del proyecto.

El proyecto tiene previsto realizar entrevistas para relevar los requerimientos; las que podrán repetirse en cada iteración según la necesidad, generando la actualización de la Especificación de Requerimientos de Software del producto (ERS). Todas las entrevistas deberán estar documentadas en minutas de relevamiento.

Como resultado de la Planificación, se generarán los siguientes documentos: Estimación, WBS, Cronograma del Proyecto (Gantt), Plan de Desarrollo de Software, Plan de Iteración y Planes de Soporte (Plan de Gestión de Riesgos, Plan de Calidad, Plan de Prueba del Proyecto, Plan de Despliegue, Plan de Aceptación del Producto, Plan de Gestión de Configuración de Software y Plan de Participación de Involucrados).

La Asociación Cordobesa de Fútbol ha solicitado que cada 15 días el Líder de Proyecto informe sobre el avance del proyecto, para lo cual deberá generar un Reporte de Control de Estado.

Las reuniones de seguimiento del proyecto deberán estar documentadas bajo las respectivas minutas de reunión, dejando constancia del tema tratado, decisiones tomadas, fecha y hora de inicio, participantes y duración de la misma. Estas minutas podrán ser enviadas por mail a todo el equipo del proyecto, por lo que resulta importante resguardar estos mails.

Propuesta de Solución

Estructura de Repositorio



Ítems de Configuración

Nombre del Ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de Ítem [Producto / Proyecto / Iteración]
Product Backlog	LRF_Product_Backlog	http://<nro IP servidor>/LRF/Proyecto/02 Product Backlog	Producto
Sprint Backlog	LRF_<Sprintxx>_Backlog	http://<nro IP servidor>/LRF/Proyecto/04 Sprints/Sprint#/01 Sprint Backlog	Producto
Métricas del Sprint	LRF_<Sprintxx>_<NombreMétrica>	http://<nro IP servidor>/LRF/Proyecto/04 Sprints/Sprint#/04 Métricas del Sprint	Iteración
Métricas del Proyecto	LRF_Métrica_<NombreMétrica>	http://<nro IP servidor>/LRF/Proyecto/03 Métricas	Proyecto
Riesgos	LRF_Riesgos	http://<nro IP servidor>/LRF/Proyecto	Proyecto
Lista de Impedimentos	LRF_Lista_Impedimentos<#sprint>	http://<nro IP servidor>/LRF/Proyecto	Proyecto
User Story	LRF_UserStory_<Nombre US>_<Nro. US>	http://<nro IP servidor>/LRF/Proyecto/01 User Stories	Producto
Revisión de Pares	Revision_Par_aaaamdd	http://<nro IP servidor>/LRF/Proyecto/04 Sprints/Sprint#/03 Revisiones	Iteración
Arquitectura	LRF_Arquitectura_NombreModelo	http://<nro IP servidor>/LRF/Producto/Arquitectura	Producto
Código fuente	LRF_<NombreClase>	http://<nro IP servidor>/LRF/Producto/Código Fuente	Producto

Sigla	Significado
<nro IP servidor>	Número de IP del servidor en el que se encuentran el repositorio del Proyecto Liga Regional de Fútbol. Ej: 172.150.1.33
<Sprintxx>	Número del Sprint. Ej: SPRINT2
<Nombre US>	Nombre de la User Story
<Nro. US>	Número que identifica la User Story
<Nombre Clase>	Nombre que identifica a cada clase implementada
<Nombre Métrica>	Nombre que identifica a la métrica

Ejercicio 4: Caso Práctico – Software para Call Center

Presentación del Caso de Estudio

Con motivo de la modernización de su Call Center, una empresa de servicios ha solicitado a una consultora de desarrollo de software la construcción de un producto destinado a facilitar la gestión de las llamadas, desde que el cliente se comunica con la empresa hasta que la llamada es finalizada. Con este nuevo producto la empresa espera además obtener mediciones que le permitan mejorar el servicio.

Para este fin la consultora conforma un scrum team para que desarrolle el producto, quiénes comenzarán a trabajar en el mismo el primero de agosto. En la grooming meeting el Product Owner (PO) le plantea al equipo que para enero del año próximo (aprovechando la menor afluencia de llamadas por el período vacacional) está previsto un importante corte del servicio por razones de mantenimiento, con lo cual para ese momento el sistema debería estar disponible para ser utilizado, mejorando así la atención a los clientes que llamen para informarse.

A continuación, se transcribe parte de la entrevista entre el Product Owner (PO) y el Scrum Team (ST):

PO: El software a construir deberá conectarse al sistema telefónico de líneas rotativas, recibir las llamadas de los clientes y derivarlos a los Operadores Telefónicos para su atención, según corresponda. Para esto deberá contar con una herramienta para generación de campañas de IVR que permita construir distintos circuitos de atención mediante la generación de “árboles de decisión y/o procesos” en forma gráfica. Para esto necesitamos una interfaz GUI (Interfaz de Usuario Grafica), que pueda ser utilizada por el Supervisor del Call Center.

ST: ¿Qué es un IVR?

PO: La respuesta de voz interactiva o IVR (del inglés Interactive Voice Response) consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como «sí», «no» u otras. Es un sistema automatizado de respuesta interactiva, orientado a entregar o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras operaciones.

ST: ¿Cómo funciona?

PO: El cliente realiza una llamada a un número de teléfono, el sistema de audio contesta la llamada y le presenta al cliente una serie de acciones a realizar, esto se hace mediante mensajes (menús de opciones) previamente grabados en archivos de audio (por ejemplo, «pulse o diga uno para ventas, dos para administración»). El cliente elige la opción a realizar introduciendo un número en el teclado del teléfono o diciendo dicho número y navega por los diferentes menús hasta encontrar la información solicitada o que el sistema dirija la llamada al destinatario elegido.

ST: ¿Qué significa que el sistema permita generar campañas de IVR?

PO: Quiere decir que las opciones que el sistema le presenta al cliente a través del IVR son configurables. Por ejemplo, ante un corte de servicio puede informarle al cliente, previa solicitud del ingreso de su número de cliente, si está afectado o no con dicho corte. Una vez finalizado el corte esa rama del árbol del IVR se deshabilita.

ST: ¿Y cuándo interviene el Operador Telefónico?

PO: Cuando el cliente no puede resolver su consulta a través del IVR es derivado a una Cola de atención, en donde lo atenderá un Operador Telefónico, que tiene asignado un número de posición.

ST: ¿Le sirve de utilidad la información solicitada por el IVR al Operador Telefónico?

PO: Si. Si se le solicita al cliente el motivo de su llamada el Operador Telefónico contará con esta información al momento de atenderlo.

ST: ¿Cómo distribuye el sistema las llamadas entre los operadores?

PO: El sistema deberá distribuir las llamadas entrantes en cola a las distintas posiciones de Operadores Telefónicos, permitiendo una carga de trabajo uniforme por operador. Esta distribución se realizará en base a grupos de atención, especializados por temas, tipos de cliente, etc., de acuerdo al perfil del Operador Telefónico (Skill). Esta distribución podrá ser cambiada dinámicamente por el Supervisor, siguiendo la evolución de los requerimientos sobre el Call Center y de acuerdo a las habilidades y conocimientos de los Operadores Telefónicos.

ST: ¿Cómo conoce el sistema que operadores telefónicos están disponibles para recibir llamadas?

PO: El sistema deberá administrar los estados del Operador Telefónico agrupados bajo la siguiente disposición:

- **Listo:** este estado se utiliza cuando el Operador Telefónico desea recibir llamadas provenientes de las colas. Optar por este estado implica conectarse a la/s cola/s que cada Operador Telefónico tenga configuradas, en base al grupo al cual pertenece y al perfil configurado por el supervisor.
- **Ocupado:** una vez que el Operador Telefónico recibe una llamada de una cola pasa automáticamente al estado Ocupado. Este estado indica que el Operador Telefónico se encuentra cursando una llamada, por lo que el sistema no cuenta con él para la transferencia de las próximas llamadas mientras se encuentre en ese estado. Se trata de un estado totalmente forzoso y automático, dado que es el sistema y no el Operador Telefónico es quien cambiará el estado de Listo a Ocupado al transferirle una llamada. En este estado el supervisor tendrá la posibilidad de monitorear la llamada en curso.
- **Libre:** este estado indica que el Operador Telefónico se encuentra dentro del ámbito de trabajo (logueado) y que no se encuentra hablando, pero que no desea recibir llamadas. Dicho estado no es forzoso, sino que es el Operador Telefónico quien decide cuando usarlo. Es útil para denotar tareas que no tengan que ver con la gestión telefónica, como la carga de datos, el envío de información por e-mail, entre otras. Equivale al deslogueo de las colas, por lo que al entrar en este estado el sistema deja de considerar a este Operador Telefónico como parte del grupo disponible.
- **Fuera:** indica que el Operador Telefónico no se ha logueado en la Plataforma, probablemente porque no se encuentra en su ámbito de trabajo. En este estado no se realiza ningún tipo de cálculo de tiempos del Operador Telefónico, pues este no se encuentra trabajando.
- **Pausas configurables:** esta funcionalidad es de suma utilidad a la hora de intentar medir cuanto tiempo le dedican los Operador Telefónicos a ciertas tareas que no tienen que ver con la gestión telefónica, ya sean parte o no del trabajo del Operador Telefónico. Ejemplos de configuración de estas pausas pueden ser: Descanso, Almuerzo, Baño, Capacitación, Consulta al supervisor o Tareas administrativas.

ST: ¿Qué ocurre si el Operador Telefónico al que el sistema deriva la llamada no atiende?

PO: En el sistema deberá ser posible determinar el tiempo máximo de espera (cantidad de Rings) en ser atendido cuando es pasada la llamada a un Operador Telefónico y no es atendida. En caso de superarse ese

tiempo, la llamada es reasignada automáticamente a otro Operador Telefónico y le cambia el estado a Libre al Operador Telefónico que no respondió.

ST: ¿Cómo monitorea el supervisor la operatoria del Call Center?

PO: El sistema deberá permitir monitorear tanto los distintos estados de un Operador Telefónico como el de todo el sistema en base a reportes en tiempo real y monitoreo en línea (semáforo), similar al que se muestra a continuación:



El sistema también deberá proveer estadísticos y reportes históricos, para un análisis posterior.

El equipo estableció que los Sprint sean de una duración de 15 días

Se considera que el éxito del sistema a construir tiene un impacto directo en la imagen de la empresa de servicios, ya que afecta directamente la calidad de atención. Asimismo, es de suma importancia asegurar la calidad desde la primera versión del producto, ya que su buen o mal funcionamiento impacta directamente en el cliente al momento de comunicarse con la empresa.

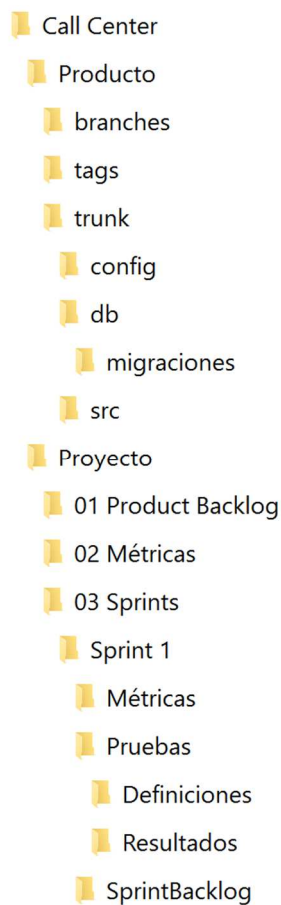
Con el objetivo de contar con diferentes configuraciones para cada entorno de trabajo (desarrollo, testing, preproducción) se guardarán los diferentes valores para cada servicio como conexión a la base de datos, configuración del servidor de email saliente, motor de caché, etc. en diferentes carpetas. Además, para mantener la sincronización entre las bases de datos de los desarrolladores el equipo ha implementado una estrategia llamada “migraciones”, con la cual se crean diferentes archivos de script incrementales ordenados por fecha que van incorporando los cambios en las tablas afectadas.

El proyecto se desarrollará bajo el framework Scrum y los requerimientos capturados en base a las entrevistas realizadas a los posibles usuarios serán recopilados en formato de User Stories.

Debido a que el equipo tiene una fecha de entrega pautada es sumamente importante tomar las métricas necesarias para realizar el correcto seguimiento del proyecto que garantice finalizar el producto en término y con una cantidad de defectos aceptable.

Propuesta de Solución:

Estructura del repositorio y las reglas de nombrado de los ítems de configuración:



Nombre del Ítem de Configuración	Regla de Nombrado	Ubicación Física	Tipo de Ítem [Producto / Proyecto / Sprint]
Product Backlog	CC_Product_Backlog.jpg	CallCenter/Proyecto/01 Product Backlog	Producto
Sprint Backlog	CC_Sprint_Backlog_<<N>>.jpg	CallCenter/Proyecto/03 Sprints/Sprint N/Sprint Backlog	Sprint
Script de migración de base de datos	CC_Migraciones_<<AAAAM MDDHHmmSS>>_<<Nombre Migración>>.<<ext>>	CallCenter/Producto/trunk/db/migraciones	Producto
Código fuente	CC_<<NombreClase>>.java	CallCenterProducto/trunk/src	Producto
Archivo de configuración	CC_Configuracion_<<ENTORNO>>_<<SERVICIO>>.cfg	CallCenterProducto/trunk/config	Producto

User Story	CC_<<N>>_US_<<Nombre_User_Story>>	CallCenter/Proyecto/03 Sprints/Sprint N/Sprint Backlog	Producto
Definición de Caso de Prueba	CC_Test_Case_<<N>>_<<Nombre>>.xlsx	CallCenter/Proyecto/03 Sprints/Sprint N/Pruebas/Definiciones	Sprint

Nota: las user stories se administran en una herramienta, por eso no tiene extensión el nombre del archivo.

Referencias:

- N es un número, de sprint, de test case o user story según corresponda.
- AAAA año, MM mes, DD día, HH hora, mm minutos, ss segundos.
- ENTORNO puede ser Desarrollo, Testing, Staging (Preproducción) o Producción.
- SERVICIO puede referirse a email, cache, database, etc.

Fuentes de Información

- Todos los casos de estudio planteados, son elaboración del equipo de Formadores que preparó el material. (Meles, Judith /Robles Joaquín / Fey Candelaria), basados en casos creados por Judith Meles, Joaquín, Robles, Cecilia Massano y Laura Covaro para la Cátedra de Ingeniería de Software, de la UTN FRC.