ENUNCIADOS DE TRABAJOS PRÁCTICOS evaluables

Ingeniería de Software

íNDICE

[PRÁCTICO 1 - SCM – Estructura de Repositorio (Entregable) 3](#_Toc489174465)

[PRÁCTICO 2 - SCM – Herramientas de SCM (Entregable) 4](#_Toc489174466)

[PRÁCTICO 3 - TESTING – Métodos de Caja Negra 5](#_Toc489174467)

[PRÁCTICO 4 - TESTING – Métodos de Caja Blanca 7](#_Toc489174468)

[PRÁCTICO 5 - TESTING – Ejecución de Casos de Prueba (Entregable) 10](#_Toc489174469)

[PRÁCTICO 6 - REVISIONES TÉCNICAS – Inspección de Código 11](#_Toc489174470)

[PRÁCTICO 7 - REVISIONES TÉCNICAS – Inspección con Herramienta (Entregable) 13](#_Toc489174471)

[PRÁCTICO 8 - REQUERIMIENTOS ÁGILES – USER STORIES 14](#_Toc489174472)

[PRÁCTICO 9 - REQUERIMIENTOS ÁGILES – USER STORIES y ESTIMACIONES 15](#_Toc489174473)

[PRÁCTICO 10 - SCRUM – Prácticas de Estimación, Planificación y Seguimiento (Entregable) 17](#_Toc489174474)

[PRÁCTICO 11 - SCRUM – MONITOREO Y MÉTRICAS ÁGILES (Entregable) 21](#_Toc489174475)

[PRÁCTICO 12 - RETROSPECTIVA (Entregable) 23](#_Toc489174476)

[PRÁCTICO 13 - LA CAJA DE PRODUCTO (Entregable) 24](#_Toc489174477)

# SCM – Estructura de Repositorio (Entregable)

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro. 2: Gestión del Software como producto** |
| **Consigna:** | Comprender los conceptos de SCM expuestos en la clase teórica, para aplicarlos en un ejemplo concreto en el contexto de la materia. |
| **Objetivo:** | Que el alumno pueda inferir cómo definir una Estructura de Repositorio e identificar los Ítems de Configuración que deberá asignar en las carpetas definidas en la estructura. |
| **Propósito:** | Familiarizarse con las actividades de SCM y repasar los conceptos de Gestión de Configuración de Software. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Plantilla de Ítems de Configuración. Bibliografía referenciada sobre el tema. |
| **Salida:** | Estructura del repositorio en formato de árbol de carpetas.  Planilla de ítems de configuración con reglas de nombrado y ubicación física. |
| **Instrucciones:** | * Pensar en todo el material que le suministra la materia y el que generará cada integrante del grupo a lo largo del cursado que necesitará ser mantenido bajo control de versiones como trabajos prácticos, resoluciones de ejercicios, entre otros. * Definir cómo sería la estructura del repositorio de la materia considerando que debe estar presente el ítem de configuración **Modalidad Académica** en la ubicación definida por la Cátedra. * Identificar cada uno de los ítems de configuración incluyendo la primera presentación de Clase Teórica cuyo archivo se denomina “01 Ingenieria de Software 2017 2C.pdf” * Para cada Ítem de Configuración determinar nombre, regla de nombrado, ubicación física y tipo (Cátedra, Clase o Producción Propia) y completar la plantilla correspondiente. |

**Plantilla de Ítems de Configuración**

| **Nombre del Ítem de Configuración** | **Regla de Nombrado** | **Ubicación Física** | **Tipo de Ítem** |
| --- | --- | --- | --- |
| **Modalidad Académica** | UTN FRC Modalidad Ing SW 2017 2C.pdf | /Reglamentacion | Cátedra |
| **Bibliografía Referenciada** | Bibliografia de Ingenieria de Software 2017.pdf |  |  |
|  |  |  |  |
|  |  |  |  |

# SCM – Herramientas de SCM (Entregable)

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro. 2: Gestión del Software como producto** |
| **Consigna:** | Comprender los conceptos de administración de configuración de software (SCM) expuestos en la clase teórica para aplicarlos en un ejercicio propuesto por la cátedra. |
| **Objetivo:** | Que el alumno sea capaz de realizar actividades básicas de la gestión de configuración mediante el uso de una herramienta tales como la definición de una estructura de repositorio, ingreso y extracción de ítems de configuración del repositorio y definición de líneas base. |
| **Propósito:** | Aplicar los conceptos de gestión de configuración estudiados en una herramienta de software específica |
| **Entradas:** | Conceptos teóricos sobre el tema desarrollados en clase. Bibliografía referenciada sobre el tema.  Estructura del Repositorio y Plantilla de Ítems de Configuración del Trabajo Práctico Evaluable 1 |
| **Salida:** | URL y Credenciales de acceso para el repositorio implementado  En la primera instancia se evaluará que:   * El repositorio sea accesible de forma pública * Se haya implementado la estructura de carpetas propuesta * Se encuentren accesibles los ítems de configuración **Modalidad Académica** y **Bibliografía Referenciada** en las ubicaciones previstas * Se encuentre disponible el archivo “*01 Ingenieria de Software 2017 2C.pdf”* en la ubicación correspondiente a su definición como Ítem de Configuración   En la segunda instancia se evaluará que:   * Cada Integrante del Grupo haya realizado operaciones de *Commit* (colocar) y *Update* (obtener) a lo largo del cursado de la materia * Los ítems de configuración disponibles presenten una evolución acorde a su contenido a lo largo del cursado con participación de los integrantes del grupo * Exista al menos una Línea Base creada en el repositorio |
| **Instrucciones:** | * Crear un repositorio de acceso público * Crear cuentas de usuario para cada uno de los integrante del grupo * Implementar la estructura del repositorio propuesta en el Trabajo Práctico Evaluable 1 * Realizar Commit (colocar) de cada ítem de configuración disponible al momento * Mantener cada uno de los ítems de configuración que se necesitarán para el cursado y aprobación de la materia bajo los niveles adecuados de administración de configuración. * Definir al menos un momento que considere adecuado para marcar una línea base y luego marcar la línea base definida en el repositorio. |
| **Observaciones:** | El repositorio implementado debe ser de acceso público utilizando Git o Subversion como motor de control de versiones. |

# TESTING – Métodos de Caja Negra

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro. 3: Aseguramiento de Calidad de Proceso y de Producto** |
| **Consigna:** | Realice el diseño de las pruebas de caja negra según los métodos de Clases de Equivalencia y Análisis de Valor límite |
| **Objetivo:** | Que el alumno pueda diseñar pruebas mediante los métodos de Clases de Equivalencia y Análisis de Valor límite, para historias de usuario dadas. |
| **Propósito:** | Familiarizarse con los métodos de Clases de Equivalencia y Análisis de Valor límite para el diseño de pruebas de caja negra. Descubrir escenarios de prueba que no se derivan a través de los métodos. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema.  Enunciado, consigna y User Story. |
| **Salida:** | Se realizará una puesta en común del conjunto de clases de equivalencia y el resultado de la aplicación del método de análisis del valor límite.  Este práctico no se entrega y por lo tanto no tiene nota. El tema se evalúa en el parcial. |
| **Instrucciones:** | Según el método de partición de equivalencia, defina las clases existentes utilizando el siguiente cuadro para la Historia de Usuario “Definir precios”.   | Condición externa | Clases de equivalencia válidas | | Clases de equivalencia inválidas | | | --- | --- | --- | --- | --- | |  |  |  |  |  | |  |  |  |  |  |   Describa la cantidad mínima de casos de prueba que requiera para asegurar la prueba de todas las clases identificadas y para asegurar que se realicen todas las pruebas de aceptación requeridas por el PO, aplicando los métodos de *Partición de Equivalencias* y de *Análisis de Valores Límite (*para los precios de las entradas*)*.  Utilice el template Template\_Casos\_De\_Prueba. |

**Festival de Folclore**

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.

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.

|  |  |
| --- | --- |
| **Definir precios**  Como Responsable de Festival quiero fijar precios de un festival para poder vender entradas a los precios fijados.  Nota: tener en cuenta que el precio depende de la noche del festival, el sector y el tipo de entrada. Todos los datos son requeridos. | **2** |
| **Pruebas de Usuario**   * Probar registrar precios para una noche, sector y tipo de entrada que ya los tiene (falla) * Probar registrar precios para una noche, sector y tipo de entrada (pasa) * Probar registrar precios para un festival vigente (pasa) * Probar registrar precios cuando no hay festival vigente (falla) * Probar registrar precios sin ingresar el precio de la entrada (falla) | |

# TESTING – Métodos de Caja Blanca

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro. 3: Aseguramiento de Calidad de Proceso y de Producto** |
| **Consigna:** | Analice el pseudocódigo según los métodos de caja blanca |
| **Objetivo:** | Que el alumno pueda realizar actividades que cubran los distintos aspectos que conforman el proceso de Testing, desde la planificación hasta ejecución de las pruebas, pasando por el diseño de los casos de prueba utilizando distintos métodos de prueba de caja. |
| **Propósito:** | Familiarizarse con los conceptos y actividades principales de los métodos de Testing de caja blanca. |
| **Entradas:** | Conceptos teóricos de Testing, desarrollados en clase. Bibliografía sobre el tema, citada.  Enunciado, consigna, templates y porción de pseudocódigo. |
| **Salida:** | Casos de prueba de caja blanca diseñados.  Este práctico no se entrega y por lo tanto no tiene nota. El tema se evalúa en el parcial. |
| **Instrucciones:** | Describa la cantidad mínima de casos de prueba que se requieran para el pseudocódigo presentado, aplicando el método de Caja Blanca.  Utilice la tabla que indica los distintos métodos de cobertura. Para cada fila, en la primera columna deberá completar la mínima cantidad de casos de prueba que deben ejecutarse y en la columna “Dato” las características de las entradas de cada una de las pruebas, como lo indican los ejemplos que se encuentran bajo dicha columna. |

**Festival de Folclore**

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.

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.

A continuación, se presenta una porción de pseudocódigo que resuelve la US de “Definir precios”:

If (Noche de festival vigente = Yes AND Sector habilitado = Yes)

If (Precio registrado para tipo Entrada = NO)

If (Precio valor numérico AND precio >= 0,00)

[Registrar precio de Entrada]

Else [Mostrar el mensaje “El precio debe ser mayor a cero”]

Precondiciones:

* Los sectores Violeta, Rojo, Verde, y Amarillo fueron registrados y están habilitados
* El sector Azul se deshabilitó
* Del viernes 22/11 al domingo 24/11 se desarrollará el Festival
* Los tipos de entradas registrados son: Menores, Estudiantes, Jubilados, General.
* Todos los precios de entradas para jubilados ya han sido registrados.

|  | **Cantidad de Casos de Prueba** | **Dato** |
| --- | --- | --- |
| **Cobertura de sentencias** |  |  |
| **Cobertura de decisión** |  | *Precio $50,00 para la noche del 22/11, sector Rojo, tipo de entrada “Estudiante”* |
| **Cobertura de condición** |  |  |
| **Cobertura de condición /decisión** |  |  |
| **Cobertura múltiple** |  | *Precio para la noche del 25/11 y el sector Azul* |

# TESTING – Ejecución de Casos de Prueba (Entregable)

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro. 3: Aseguramiento de Calidad de Proceso y de Producto** |
| **Consigna:** | Ejecute los casos de pruebas de caja negra diseñados |
| **Objetivo:** | Que el alumno pueda aplicar los distintos aspectos del testing, desde la planificación hasta ejecución de las pruebas. |
| **Propósito:** | Comprender como se realiza la ejecución de los casos de prueba diseñados |
| **Entradas:** | Conceptos teóricos sobre el tema desarrollados en clase. Bibliografía referenciada sobre el tema.  Planilla de Excel con User Story, Plantilla de Casos de Prueba y Reporte de Defectos  MercadoAbasto\_Template\_Caso\_De\_Prueba\_Defectos.xlsx |
| **Salida:** | Planilla de Excel con Clases de Equivalencias identificadas  Tres Casos de Prueba diseñados  Ejecución y Reporte de Defectos |
| **Instrucciones:** | * En base a la User Story “Registrar Alquiler de Puesto” identificar todas las clases de equivalencia * Diseñar tres Casos de Prueba utilizando las estrategias de Clases de Equivalencia y Análisis de Valores Límite para encontrar al menos tres Defectos * Ejecutar los Casos de Prueba antes diseñados y registrar los Defectos encontrados completando la Plantilla correspondiente |
| **Observaciones:** | La funcionalidad implementada está disponible en <https://drive.google.com/drive/folders/0B3qJ3d7wA0lSaUY1ZWc1N3pQbms?usp=sharing>.  Para ejecutar el software se requiere Java 8 que se puede descargar desde <https://www.java.com/es/download>. |

# REVISIONES TÉCNICAS – Inspección de Código

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro. 3: Aseguramiento de Calidad de Proceso y de Producto** |
| **Consigna:** | Ejecute una revisión técnica a la porción de código entregada y genere el reporte correspondiente. |
| **Objetivo:** | Qué el alumno pueda ejecutar un proceso formal de inspección de código, identificando hallazgos en una porción de código. |
| **Propósito:** | Familiarizarse con proceso formal de inspección de código. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Estándar de Inspección de Código. Bibliografía referenciada sobre el tema.  Paper 11\_Best\_Practices\_For\_Peer\_Code\_Review\_Final.  Enunciado, consigna y porción de código. Estándar de Reporte de Inspección. |
| **Salida:** | Hallazgos identificados.  Este práctico no se entrega y por lo tanto no tiene nota. |
| **Instrucciones:** | * Lea y analice la siguiente porción de código. * Realice una inspección del código mediante el método formal de inspección y prepara el reporte de inspección correspondiente respetando el estándar. |

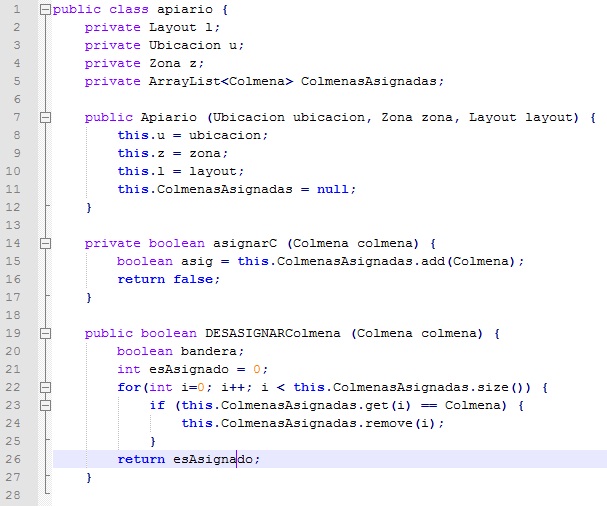
**Monitoreo de Colmenas**

“Miel Serrana” es un grupo de apicultores que se dedican a la producción de miel y otros productos que se obtienen de las abejas. Con el objeto del monitoreo de las colmenas, han solicitado un sistema web que cumpla con este propósito. Se ha solicitado a una consultora de la ciudad de Córdoba que sea la responsable del desarrollo del producto.

El proyecto mediante el cual se desarrollará el producto solicitado es denominado por sus integrantes “iBee”. La consultora se propuso para esta instancia mejorar el tiempo de entrega del producto al cliente.

El conjunto de colmenas que un apicultor tiene en un determinado lugar físico o zona se denomina apiario. Para cada apiario se debe registrar la fauna de la zona, la flora de la zona y el clima. Para el registro del clima se debe tener en cuenta temperatura, humedad, índice UV y viento. La actualización permanente es muy importante, debido a que para cada atributo que se mide del mismo se definen alarmas que deben dispararse automáticamente en la estación central de cada apiario. El sistema de alarmas debe ser cuidadosamente testeado, ya que una falla en este módulo puede hacer que el producto fracase. La consultora se propuso disminuir la densidad de defectos encontrados en el testing.

La consultora tiene definidas reglas dentro de su proceso para la escritura del código, a las que iBee debe adaptarse. Entre ellas, utilizar notación camelCase para los nombres de clases, métodos y atributos, y utilizar nombres de variables que sean representativas de lo que significan. Además, dentro de la consultora son muy exigentes con los comentarios dentro del código.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Reporte de Inspección** | | | | | | | |
| **Fecha de Inicio:** |  | |  | | | | |
| **Revisores:** | **Nombre** | | |  | **Rol** | | |
| ISW | | |  | Autor | | |
| Silva, Federico | | |  | Moderador | | |
| Brond, Matías | | |  | Inspector | | |
| Michellini, Guido | | |  | Lector | | |
|  | | |  |  | | |
| **Alcance de la revisión:** |  | | Porción de código de la clase “apiario”, de la línea 1 a 27. | | | | |
|  | | | | | | | |
| **Nombre** | **Versión** | | |  | **Autor** | | |
| Clase Apiario | Initial Version (ó 1.0) | | |  | @ISW | | |
|  |  | | |  |  | | |
|  |  | | |  |  | | |
| **Hallazgos** | | | | | | | |
| **Nº** | **Fecha** | **Descripción** | | | | **Línea** | **Severidad** |
| 1 | 19/09/2017 | El nombre de la clase no respeta la notación camelCase. | | | | 1 | Menor |
| 2 | 19/09/2017 | La variable “l” no es representativa. | | | | 2 | Menor |
| 3 | 19/09/2017 | La variable “u” no es representativa. | | | | 3 | Menor |
| 4 | 19/09/2017 | La variable “z” no es representativa. | | | | 4 | Menor |
| 5 | 19/09/2017 | El arrayList colmenasAsignadas es “NULL” por lo tanto no pueden efectuarse operaciones. | | | | 11 | Mayor |
| 6 | 19/09/2017 | El nombre del método asignarC no es representativo. | | | | 14 | Menor |
| 7 | 19/09/2017 | El método asignarC siempre retorna el valor “FALSE” | | | | 15 | Menor |
| 8 | 19/09/2017 | El nombre del método DESASIGNARcolmena no respeta la notación camelCase. | | | | 19 | Menor |
| 9 | 19/09/2017 | La variable “bandera” no es representativa. | | | | 20 | Menor |
| 10 | 19/09/2017 | La variable “bandera” está declarada pero no es utilizada. | | | | 20 | Menor |
| 11 | 19/09/2017 | La variable “esAsignado” está declarada pero no es utilizada. | | | | 21 | Menor |
| 12 | 19/09/2017 | El tipo de dato de la variable “esAsignado” no corresponde con el tipo de dato del retorno del método. | | | | 21 | Mayor |
| 13 | 19/09/2017 | El nombre del constructor no es el mismo que el de la clase. | | | | 7 | Menor |
| 14 | 19/09/2017 | La cabecera del ciclo “for” no está formulada correctamente. | | | | 22 | Mayor |
| 15 | 19/09/2017 | Existe una comparación entre una clase y un objeto. | | | | 23 | Mayor |
| 16 | 19/09/2017 | Falta cierre de la llave del ciclo “for”. | | | | 22 | Mayor |
| 17 | 19/09/2017 | No se puede añadir la clase “Colmena” en una arrayList. | | | | 15 | Mayor |
| 18 | 19/09/2017 | Faltan comentarios en general en toda la porción de código. | | | | N/A | Menor |
| 19 | 19/09/2017 | El modificador de acceso del método “asignarC” es incorrecto. | | | | 14 | Mayor |
| 20 | 19/09/2017 | Faltan métodos de seteo en la clase. | | | | N/A | Menor |
| 21 | 19/09/2017 | La variable “asig” no es representativa. | | | | 15 | Menor |
| **Decisión Final:** |  | | Aceptar/***Rechazar***/Aceptar Provisoriamente | | | | |

# REVISIONES TÉCNICAS – con soporte de Herramienta de software (Entregable)

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro. 3: Aseguramiento de Calidad de Proceso y de Producto** |
| **Consigna:** | Ejecute una revisión técnica a la porción de código entregada utilizando una herramienta de análisis estático de código y genere el reporte correspondiente. |
| **Objetivo:** | Qué el alumno pueda realizar una revisión técnica utilizando una herramienta de software de soporte, que le permita identificar hallazgos en una porción de código. |
| **Propósito:** | Familiarizarse con el uso de Revisiones Técnicas de Software utilizando herramientas de software. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema.  Enunciado, consigna y porción de código. |
| **Salida:** | Reporte de Hallazgos identificados por la herramienta. |
| **Instrucciones:** | * Lea y analice la porción de código entregada * En base a las características técnicas del artefacto de código seleccione una herramienta de análisis estático * Realice una Revisión Técnica sobre el código entregado utilizando la herramienta seleccionada * Genere el Reporte de Hallazgos y preséntelo en formato PDF |

# REQUERIMIENTOS ÁGILES – USER STORIES

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro.4: Gestión Ágil de Proyectos** |
| **Consigna:** | Identificar y escribir las US identificadas en el Dominio presentado para el práctico |
| **Objetivo:** | Aplicar los conceptos teóricos desarrollados en clase sobre User Stories. |
| **Propósito:** | Familiarizarse con los conceptos de requerimientos ágiles y en particular con US, Épicas y Temas. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Filminas.  Enunciado, consigna y tarjeta de US. |
| **Salida:** | US identificadas y tarjetas de US completas.  Este práctico no se entrega y por lo tanto no tiene nota. El tema se evalúa en el parcial. |
| **Instrucciones:** | Los docentes representarán a expertos del dominio que expresarán sus necesidades vinculadas al desarrollar un software para el manejo de una bomba de insulina.  Los alumnos realizarán preguntas con el objetivo de acordar juntos el alcance del proyecto, y determinar las user stories correspondientes y definir las pruebas de aceptación de usuario para cada una.  Cada grupo presentará el resultado obtenido al final de la clase al resto del curso. |

**Sistema de bomba de insulina**

***Objetivo***: Desarrollar un software para el manejo de una bomba de insulina.

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

**Product Owner (PO)**: ¿Qué tamaño tiene la bomba de insulina que vamos a fabricar? ¿Cómo se colocará la bomba de insulina?

**Experto en el Dominio (ED)**: Estamos trabajando en una bomba del tamaño de una billetera. Algo como lo que ves en esta fotografía:



La bomba de insulina **(BI)** deberá colocarse en la cintura para que inyecte insulina cuando corresponda.

**PO**: ¿Cómo debería funcionar para administrar la insulina necesaria a un diabético?

**ED**: Se debe poder configurar la insulina basal, es decir la insulina que se debe inyectar constantemente. Para esto se definen períodos de tiempo, cada cuánto se debe inyectar, y la cantidad de insulina que se debe inyectar. Por ej., para los días hábiles de 8 a 12 se debe inyectar 0.625 mg por decilitros de insulina cada 1 hora. Esto generalmente lo define y configura el médico de cabecera del paciente que usará la BI. Por otro lado, se debe definir la cantidad de insulina por carbohidratos para el bolo. El bolo es la insulina que se inyecta cuando se está por realizar alguna acción que requiere de una mayor cantidad de insulina, por ej, al momento de comer. El bolo debe ser inyectado únicamente cuando el paciente lo indique.

**PO**: ¿Y qué información deberíamos requerir para definir la cantidad de insulina del bolo?

**ED**: La cantidad de carbohidratos a ingerir y la medida actual de insulina.

**PO**:¿Alguna otra funcionalidad que deberíamos tener en cuenta? ¿Alarmas? ¿Algún cálculo extra?

**ED:** Debería poder calcularse la vida de la insulina inyectada, para conocer en este momento cuánto falta para inyectar insulina nueva. Además, deberían sonar alarmas cuando pase un período de tiempo configurado en que la bomba no está colocada en el paciente, y cuando se pase un momento de tiempo en que el paciente debería haberse inyectado una cantidad de insulina de bolo. Por último, sería útil contar con una funcionalidad para cuando el paciente indique se van a ingerir grasas o proteínas, para que la inyección de insulina se realice de forma más racionalizada.

**PO**: Respecto a la batería, ¿Deberíamos informar la cantidad de batería que le queda a la BI verdad?

**ED**: Sí, deberíamos mostrar constantemente una pila con tres niveles de carga, y al llegar al último tercio de carga la pila debería comenzar a titilar y emitir una alerta para que el paciente ponga a cargar la BI.

# REQUERIMIENTOS ÁGILES – USER STORIES y ESTIMACIONES

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro.4: Gestión Ágil de Proyectos** |
| **Consigna:** | Identificar y escribir las US identificadas en el Dominio presentado para el práctico |
| **Objetivo:** | Aplicar los conceptos teóricos desarrollados en clase sobre User Stories. |
| **Propósito:** | Familiarizarse con los conceptos de requerimientos ágiles y en particular con US, Épicas y Temas. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema.  Enunciado, consigna y tarjeta de US. |
| **Salida:** | US identificadas y tarjetas de US completas.  Este práctico no se entrega y por lo tanto no tiene nota. El tema se evalúa en el parcial. |
| **Instrucciones:** | Los docentes representarán a expertos del dominio que expresarán sus necesidades vinculadas al desarrollar un software para el seguimiento de taxis.  Los alumnos realizarán preguntas con el objetivo de acordar juntos el alcance del proyecto, y determinar las user stories correspondientes y definir las pruebas de aceptación de usuario para cada una.  Cada grupo presentará el resultado obtenido al final de la clase al resto del curso. |

**Taxi Mobile: Sistema web mobile para seguimiento de taxis**

***Objetivo:*** Desarrollar un producto de software para Smart phones, donde los pasajeros de taxis puedan solicitar el taxi más cercando 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:

|  |  |
| --- | --- |
| Prototipo para Taxista y para Pasajero | Prototipo para Central de Taxis |
| https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcR5cHKcLvUfUWeQikr-IPw6PAwWDeAMpUOsYdxwuz4_F0ngiGy7aQ | https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcR7O3Fo3QXt1dfFy-6U6f9RT8PWMwAoPR4F-Tky8X9GIuDqzklo |

# SCRUM – Prácticas de Estimación, Planificación y Seguimiento (Entregable)

|  |  |
| --- | --- |
| **Unidad**: | **Unidad Nro.4: Gestión Ágil de Proyectos** |
| **Consigna:** | A través de un juego y de las instrucciones comunicadas por los docentes, implementar las siguientes prácticas de Scrum:   * Planificación de Sprint * Ejecución de Sprint * Daily Scrum Meeting * Sprint Review * Sprint Retrospective |
| **Objetivo:** | Aplicar las prácticas de estimación, planificación y seguimiento de Scrum en la ejecución de un caso práctico. |
| **Propósito:** | Familiarizarse con las prácticas de Scrum a través de una dinámica en forma de juego. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema.  Instrucciones del juego.  Materiales necesarios. |
| **Salida:** | Mostrar los resultados de la dinámica en cada Sprint Review.  Entregar al docente   * Product Backlog con priorización de las US * Sprint Backlog de cada Sprint (incluyendo estimaciones) * Gráficos de los burndown chart de cada Sprint * Resultados de las Retrospective * Hoja final de conclusiones con respuesta a las preguntas planteadas. |
| **Instrucciones:** | El docente que dirige la dinámica asumirá el rol del Product Owner y será quién indique la consigna y el alcance de la dinámica a realizar. |

### **Proyecto: Instrumentos musicales para niños**

**Objetivo**: Fabricar prototipos de instrumentos musicales para una importante Juguetería de Córdoba. La empresa desea adquirir instrumentos musicales para niños, para lo cual ha solicitado una serie de prototipos de distintos instrumentos, lo que permitirá la atracción de más niños y potenciales clientes.

Los dueños de la Juguetería desean ponerse en contacto con distintas fábricas de estos instrumentos para comprobar sus propuestas, ya que la idea está muy orientada a la calidad y no se va a aceptar cualquier instrumento.

**Materiales que cada grupo deberá tener para el desarrollo de la clase:**

* 1 caja vacía y pequeña (tamaño deseado, para la guitarra)
* Banditas elásticas (aproximadamente 20), tamaño acorde a la caja
* Una lata vacía y limpia y sin bordes filosos, tamaño deseado (leche Nido, Pringles, etc.)
* 1 globo
* 2 vasitos de plástico
* Porotos, arroz, cualquier semilla pequeña, botones o piedritas
* 2 palitos de madera (palitos de helado)
* Una tira de papel del tamaño de los palitos de madera
* Un paquete de palillos de dientes
* 3 conos de papel higiénico o 2 de papel de cocina
* Un trozo de cartón delgado o cartulina
* 8 sorbetes o pajitas
* Tijeras - Marcadores de colores - Cinta adhesiva - Goma de pegar

**Reglas**:

* Cuando finaliza un sprint se contarán cuántos prototipos de instrumentos musicales están terminados y listos para ser entregados a la Juguetería. Sólo se aceptarán aquellos prototipos que cumplan con el 100% de lo acordado. Si un prototipo no pasa por la revisión final de calidad no se aceptará. Los prototipos sin terminar se pueden continuar en la siguiente iteración para ser completados.
* Cada equipo representará a una fábrica. Cada equipo tendrá un logotipo, que será el número de grupo que le corresponde.

**Theme**: Como Juguetería, deseamos ampliar nuestras ofertas de juguetes para niños y ser reconocidos como la mejor juguetería de Córdoba y con la más amplia variedad.

**Epic**: Contar con diferentes instrumentos musicales para niños.

|  |  |  |
| --- | --- | --- |
| **ID** | User Stories | **Valor de Negocio** |
| **1** | Yo como dueño de la juguetería quiero contar con guitarras, a fin de ofrecer el instrumento de cuerda más divertido para los chicos | **15** |
| **2** | Yo como dueño de la juguetería quiero contar con tambores de lata, a fin de ofrecer instrumentos de percusión en mi empresa | **10** |
| **3** | Yo como dueño de la juguetería quiero contar con maracas, a fin de ofrecer instrumentos idiófonos en mi empresa | **20** |
| **4** | Yo como dueño de la juguetería quiero contar con armónicas, a fin de ampliar la variedad de instrumentos musicales ofrecidos | **18** |
| **5** | Yo como dueño de la juguetería quiero contar con palos de lluvia, a fin de ofrecer instrumentos que generan sonidos ambientales | **17** |
| **6** | Yo como dueño de la juguetería quiero contar con zampoñas, a fin de ampliar la variedad de instrumentos musicales ofrecidos | **20** |

**Imágenes (a modo ilustrativo) de los prototipos deseados**

|  |  |  |
| --- | --- | --- |
| **1** | **2** | **3** |
| **4** | **5** | **6** |

**Detalle de historias de usuario.**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID de la historia** | **Actividades (orientativas)** | **Esfuerzo (minutos)** | **Puntos de historia** |
| **1** | Hacer un agujero a la caja | **2** |  |
| Atravesar las bandas elásticas de una punta a la otra, cuidando que pasen por encima del agujero, y sujetarlas | **1** |
| Personalizar prototipo | **3** |
| **2** | En la boca superior de la lata extender el látex del globo para cubrirla | **1** |  |
| Sujetar con una bandita elásticas el látex del globo | **1** |
| Personalizar prototipo | **3** |
| **3** | Colocar dentro de los vasitos los porotos, arroz, botones y/o piedritas | **1** |  |
| Pegar con cinta el otro vaso, juntando boca con boca | **1** |
| Personalizar prototipo | **3** |
| **4** | Colocar el papel entre los dos palos de madera | **1** |  |
| Fijar banda elástica en cada extremo de los palos | **1** |
| En cada extremo, introducir un palillo por encima y otro por debajo del papel, situados justo al lado de las bandas elásticas | **2** |
| Personalizar prototipo | **3** |
| **5** | Unir con cinta los conos | **1** |  |
| Atravesarlos por todos lados con palillos de dientes, formando cruces. | **4** |
| Cortar las puntas salientes de los palillos | **2** |
| Pegar o encintar cada palillo para que queden en su lugar | **2** |
| Depositar arroz dentro del cono | **1** |
| Tapar ambos extremos del cono con el cartón | **2** |
| Personalizar prototipo | **3** |
| **6** | Cortar dos pequeñas piezas rectangulares de cartón | **1** |  |
| Colocar y pegar los sorbitos en una de las piezas de cartón | **2** |
| Pegar la segunda pieza de cartón sobre la primera, de modo tal que las pajitas queden entre los dos cartones | **1** |
| Recortar la parte inferior de las pajitas, cada una debe ser más corta que la anterior | **1** |
| Personalizar prototipo | **3** |

**Conclusiones**

* ¿Se consiguió lo que el cliente quería desde el principio?
* ¿Las estimaciones han ido mejorando con el tiempo? ¿Los equipos encontraron su límite de producción? Repasar los Burndown chart.
* ¿El concepto de la retrospectiva fue útil?
* ¿Los motivó el hecho de trabajar en equipo?

# SCRUM – MONITOREO Y MÉTRICAS ÁGILES (Entregable)

|  |  |
| --- | --- |
| **Unidad**: | **Unidad Nro.4: Gestión Ágil de Proyectos** |
| **Consigna:** | En función de las User Story y estimaciones entregadas, implementar las prácticas de Scrum, haciendo foco en la definición y análisis de métricas. |
| **Objetivo:** | Aplicar las prácticas de métricas ágiles en la ejecución de un caso práctico de Scrum. |
| **Propósito:** | Repasar los conceptos vinculados a la definición de Métricas de Software.  Definir y especificar métricas ágiles de software.  Analizar e interpretar métricas ágiles de software. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema.  Enunciado del proyecto  Lista de US |
| **Salida:** | Cada grupo presentará las métricas definidas y los resultados obtenidos luego de la ejecución de cada Sprint, junto con un diagnóstico de lo ocurrido, así como también el producto final obtenido |
| **Instrucciones:** | * Dividirse en grupos * Estimar cada una de las US enunciadas * Priorizar y planificar los Sprint * Definir las métricas ágiles a utilizar * Ejecutar los Sprint (de 15 minutos cada uno), incluyendo * 2 Daily Scrum Meeting * 1 Sprint Review, registrando los resultados para cada una de las métricas definidas y un diagnóstico de la situación * Presentar el trabajo realizado   ***Tener en cuenta que:***   * Cada sprint dura 15 minutos, formados por 3 días de 5 minutos cada uno * Cada sprint planning dura 3 minutos |

**Expendedor de Productos Marinos:**

El Cliente, una marca deportiva reconocida mundialmente, quiere lanzar un producto para deportes marinos inexistente en el mercado. Con el fin de dar una mayor imagen de innovación el producto se venderá a través de un expendedor automático.

La visión del proyecto será construir un expendedor automático de un producto deportivo.

Las herramientas son las siguientes:

* + 1 afiche blanco, un marcador negro, cinta adhesiva y post it (para la planificación de los Sprint)
  + 3 cartulinas (2 blancas y 1 negra)
  + 2 tijeras
  + 1 cinta adhesiva
  + 1 regla
  + 4 globos
  + 4 bolígrafos
  + 1 instrucción para hacer un barco de papel
  + 1 revista de moda
  + Papel de diario (recortes simulando dinero)

En la siguiente tabla se describe para cada uno, el detalle de la US.

| id | Requisitos Iniciales | Valor | División de tareas | Story points |
| --- | --- | --- | --- | --- |
| 1 | Marca del Cliente: El consumidor podrá ver con claridad la marca deportiva y una foto o dibujo de temática marina, desde una distancia de tres metros | 10 | Cortar las tiras blancas (2 min)  Pegar las tiras (1 min)  Buscar la foto temática (3 min)  Pegar la la foto (1 min) |  |
| 2 | Producto: el consumidor podrá obtener el producto de una cajita especifica | 9 | Cortar la caja (5 min)  Armar la caja (5 min)  Pegar la caja (2 min) |  |
| 3 | Alcancía: el consumidor podrá pagar el producto depositando un billete en una cajita específica. | 8 | Cortar la caja (5 min)  Armar la caja (5 min)  Pegar la caja (2 min) |  |
| 4 | Transporte: un transportista, por ejemplo el Cliente, podrá trasladar el expendedor al lugar de venta, una mesa que diste tres metros, sin que se rompa. | 7 | Mover lo armado para validar si se cumple con el requerimiento (3 min) |  |
| 5 | Globos: El consumidor verá en cada uno de los dos laterales, 1 globo (2 en total), para dar una imagen de diversión del deporte náutico. | 3 | Inflar los globos (3 min)  Pegar los globos (2 min) |  |
| 6 | Barcos: el consumidor verá en los dos laterales del expendedor 3 barcos de papel (6 en total), para reforzar la imagen de deporte marino y marketing de próximos productos | 10 | Cortar la cartulina (2 min)  Armar los barcos (8 min)  Pegar los barcos (1 min) |  |

# RETROSPECTIVA (Entregable)

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro.4: Gestión Ágil de Proyectos** |
| **Consigna:** | En función de los conceptos desarrollados en la materia sobre las ceremonias de Scrum y en particular sobre Retrospectivas, aplicar la técnica al trabajo grupal realizado por los estudiantes durante el cursado de la materia Ingeniería de Software. |
| **Objetivo:** | Utilizar la técnica de retrospectiva propuesta por SCRUM para realizar una autoevaluación del grupo de trabajo en el contexto del cursado de la materia Ingeniería de Software. |
| **Propósito:** | Familiarizarse con el uso de la técnica de Retrospectivas Ágiles. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema.  Afiches  Post it’s  Sharpies  Marcadores para pizarra blanca y una pizzarra |
| **Salida:** | Conclusiones obtenidas al finalizar la retrospectiva realizada por cada grupo y las conclusiones generales del grupo clase. |
| **Instrucciones:** | Trabajará cada grupo tal como estuvieron conformados a lo largo del cuatrimestre.  Deberán realizar una retrospectiva respectando la estructura propuesta.  Obtener las conclusiones en cada grupo.  Debatir las conclusiones con el grupo clase. |

# LA CAJA DE PRODUCTO (Entregable)

|  |  |
| --- | --- |
| **Unidad:** | **Unidad Nro.4: Gestión Ágil de Proyectos** |
| **Consigna:** | Consigna fundamental: **“Un producto = una Visión = Una Caja con la Visión/Producto”** |
| **Objetivo:** | Aplicar los conceptos teóricos desarrollados en clase sobre gestión de requerimientos en ambientes ágiles. |
| **Propósito:** | Afianzar los conceptos de Requerimientos Ágiles y en particular los de User Stories, Épicas y Temas. |
| **Entradas:** | Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema..  Materiales para trabajar  Enunciado, consigna y tarjeta de US. |
| **Salida:** | Cada grupo presentará el resultado obtenido al final de la clase al resto del curso. |
| **Instrucciones:** | * Cada grupo deberá elegir un producto de software y crear, en 10 minutos, una caja que contenga la visión del mismo. * La crítica – Feedback (5 minutos): Un representante de cada grupo toma su “caja” y visita otro grupo, expone sus argumentos y escucha y anota lo que dicen los compañeros. * El grupo determina el MVP (Minimum Viable Product: es la versión mínima del producto que me permite salir a probar al Mercado y obtener feedback). |

**Materiales para trabajar**:

* 2 afiches blancos
* Fibras y marcadores de colores
* Recortes de revistas
* Papeles de colores
* Goma de pegar
* Tijeras

**Lineamientos para la Caja del Producto:**

* Frente de la Caja: Nombre, Foto – Ícono o Dibujo, Slogan, 3 o 4 argumentos de venta
* Contra frente: Vista detallada de la funcionalidad – Pre requisitos