Integración de Metodologías Ágiles en el Desarrollo de un Sistema de Monitoreo Inalámbrico para Medir la Contaminación del Aire en Tiempo Real.

Walter Fuertes, Diego Carrera, César Villacís, Fernando Galárraga, Theofilos Toulkeridis, Hernán Aules

Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas ESPE, Av. General Rumiñahui S/N,

Sangolquí, Ecuador

Contenido

- **Motivaciòn y Objetivos**
- **Contribuciòn y beneficios**
- **Marco conceptual**
- **Configuraciòn del experimento**
  - Proceso de desarrollo
- **Evaluaciòn de resultados**
- **Conclusiones**
- **Trabajo futuro**

Motivaciòn y Objetivos

- **Motivaciòn**
  - La contaminación atmosférica y la falta de puntos de monitoreo de calidad del aire son **retos ambientales y tecnológicos** de las ciudades del mundo;
  - Lamentablemente, los productos existentes y los resultados que generan, no representan **soluciones de bajo costo**;
  - Aún no se ha logrado una solución integral, debido a que estas soluciones, se han enfocado en la construcción del prototipo sin establecer un **proceso de Ingeniería de Software** que se comprometa con la calidad de software.

Motivaciòn y Objetivos

- **Objetivos**
  - El presente proyecto propone **desarrollar un sistema de monitoreo inalámbrico** de bajo costo;
  - Hemos utilizado un dispositivo electrónico, basado en **Arduino**, que está equipado con tres sensores que miden la concentración de monóxido de carbono (CO), dióxido de carbono (CO_2) y densidad de polvo;
  - Transferir inalámbricamente la información recolectada en **tiempo real**, almacenarlos en una base de datos y visualizar dicha información en una aplicación Web.

Contribuciòn y beneficios

- **Contribuciòn**

(1)Los datos generados, utilizando este prototipo, pueden también aplicarse mundialmente para **fines científicos** como son la seguridad en expediciones espeleológicas, para el control de las actividades industriales y en otras múltiples ramas científicas;

(2)La principal contribución de este proyecto es haber **integrado dos metodologías agiles** de desarrollo de software como Scrum y Extreme Programming.

Contribuciòn y beneficios

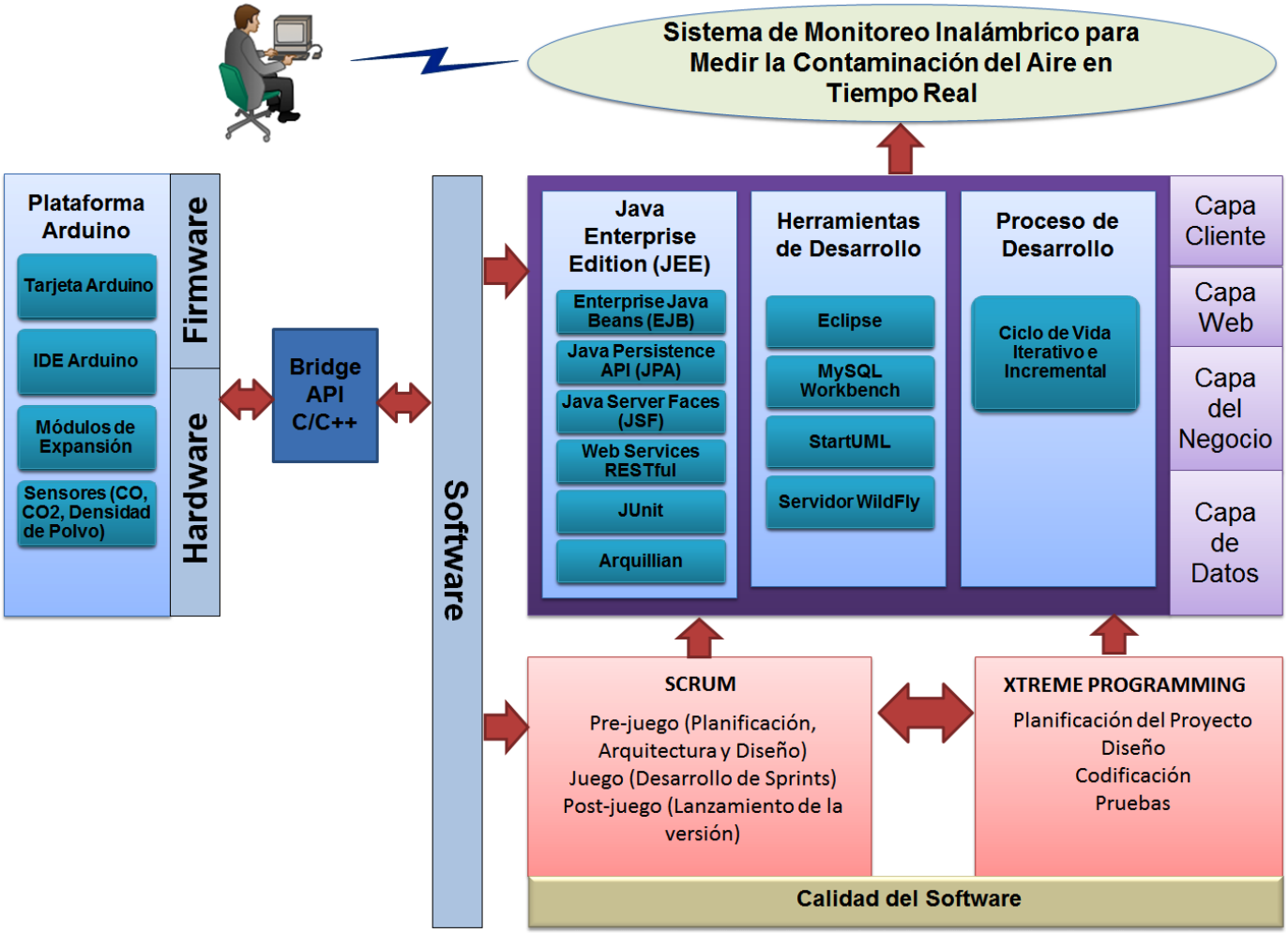
- **Beneficios**

(1) Capacidad de **adaptación a los cambios de requerimientos** del software en todo el ciclo de vida del proyecto;

(2) Facilita **períodos de desarrollo cortos** que permiten entregas de productos de software totalmente funcionales para que puedan ser probados por el cliente;

(3) El proceso de desarrollo se basó en el **ciclo de vida interactivo e incremental**, lo que permite el incremento de sensores, evitando grandes pérdidas en cuanto a costos, innovación y tiempo.

Marco conceptual



Marco conceptual

- **Scrum**
  - El uso de metodologías ágiles de desarrollo de software se ha popularizado en la última década debido a la evolución rápida y constante de la **industria del software**;
  - La **ingeniería de software ágil** representa una alternativa razonable a la ingeniería de software convencional para ciertas clases de software y en algunos tipos de proyectos y así mismo, se ha demostrado que concluye con rapidez sistemas exitosos;
  - Scrum es un método utilizado en proyectos que ayuda con la efectividad para trabajar en equipo, con **tiempos de entrega** muy cortos.

Marco conceptual

- **Extreme Programming**
  - XP, por su parte, es un método utilizado para el desarrollo de software con **requerimientos cambiantes** o vagamente definidos y en donde existe un alto riesgo técnico;
  - En esta investigación, se eligieron estas dos metodologías ágiles debido a que Scrum fue utilizada para establecer el conjunto de tareas entre la **línea base de análisis y la línea base del producto**;
  - Como un apropiado complemento, se aplicó la metodología XP para la **codificación y pruebas** de los productos incrementales desarrollados en cada iteración (sprint), englobando así todo el ciclo de vida del proyecto.

Marco conceptual

- **Integraciòn de metodologìas**
  - Para que el equipo de desarrollo realice el control y seguimiento del trabajo que se realizó, Scrum define una serie de **artefactos**; cada tarea deberá estar priorizada y debidamente estimada en relación al tiempo que se necesitará para completarla, por esta razón, el diagrama conocido como *Burn-Down Chart* establece la evolución del trabajo durante un Sprint.

  - Por otra parte, la metodología XP define la manera en la que el equipo de desarrollo trabaja, para lo cual se especifican las **prácticas**.

Marco conceptual

La tabla describe las **tareas ejecutadas** en la línea base de análisis y la línea base del producto

	Artefactos de Scrum	**Prácticas de XP**
Línea base de Análisis	Product Backlog Sprint Backlog	Diseño Incremental
Línea base del Producto	Product Increment	Pruebas unitarias Integración Continua

Tabla. Integración de las metodologías ágiles Scrum y XP.

Marco conceptual

- **Hardware y Software**
  - Se utilizó la placa Arduino, que es una **plataforma abierta** de computación compuesta por un microcontrolador montado en una placa electrónica, que puede ser usada para desarrollar objetos interactivos capaces de recolectar datos por medio de sensores y controlar luces, motores u otros elementos físicos;
  - Para la transferencia de datos entre el hardware y la aplicación Web se desarrolló un **componente de software** mediante una API desarrollada en C/C++ y que se encuentra almacenada en la memoria Flash interna de la placa Arduino.

Configuraciòn del experimento

- **Proceso de desarrollo**
  - **Scrum** define dos etapas en su proceso de desarrollo: La primera denominada el Sprint 0 o etapa inicial; y la segunda de iteraciones sucesivas, también llamadas Sprints;
  - La etapa **Sprint 0**, de duración variable pero no indefinida, determinó la viabilidad del proyecto mediante el estudio de las condiciones y recursos necesarios;
  - También se realizó la primera versión del listado de requerimientos generales del software, listado que se convirtió en el ***Product Backlog***. Una vez organizado y priorizado éste, se inició la etapa de iteraciones sucesivas o Sprints.

Configuraciòn del experimento

- **Proceso de desarrollo**
  - Los miembros del equipo de desarrollo escogieron un grupo de historias de usuario del *Product Backlog* para desarrollarlo en una sola iteración, donde a este nuevo listado de tareas se lo conoce como pila de tareas o ***Sprint Backlog***;
  - El experimento contempló tres iteraciones en la que se liberó partes del producto, ***Product Increment,*** que se inspeccionaron y se evaluaron para aumentar la funcionalidad y mejorar en calidad respecto a la anterior versión.

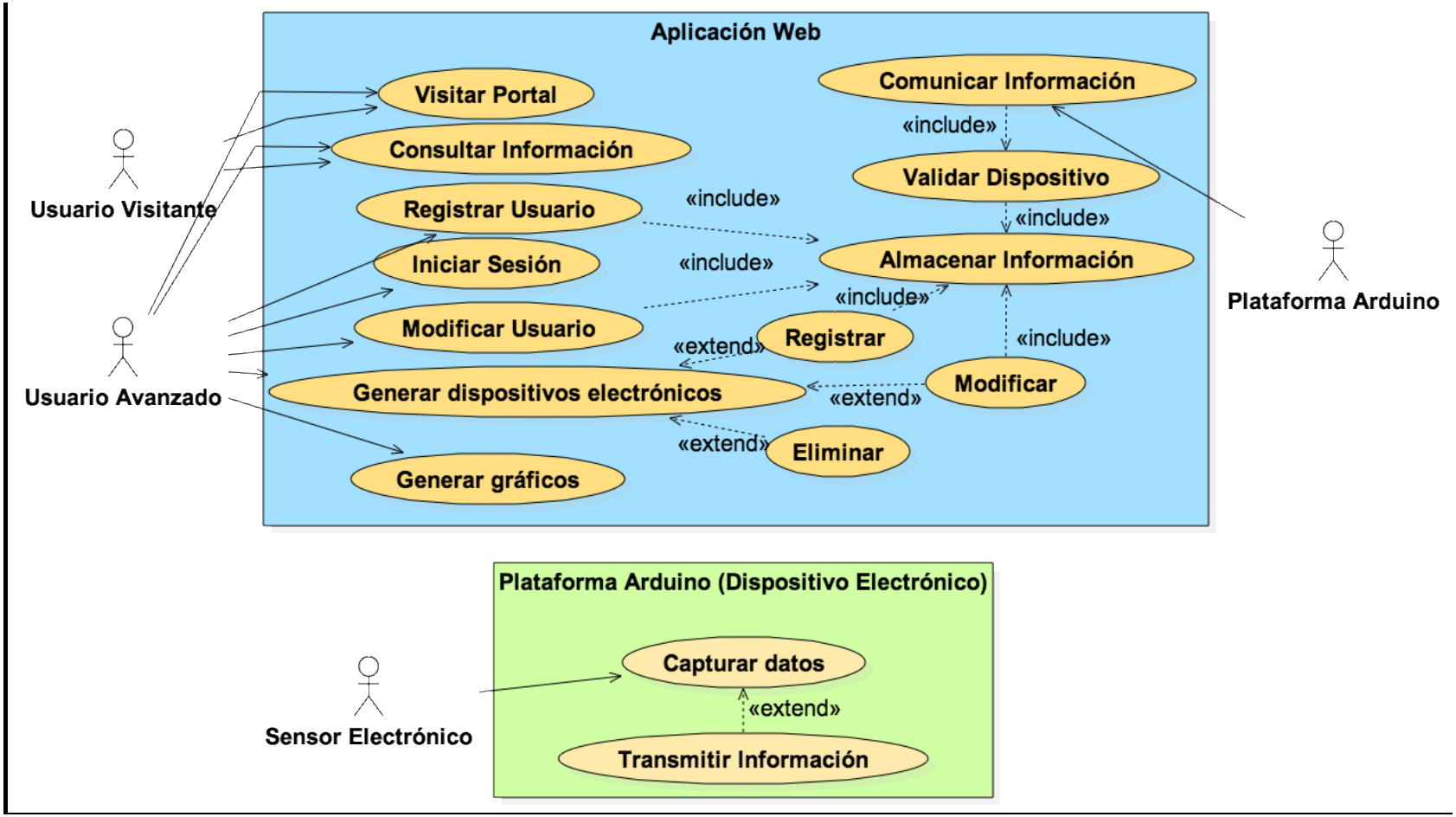
Configuraciòn del experimento

- **Proceso de desarrollo**
  - (1) Recolección e **intercambio de información**, basado en hardware, y que organiza e interpreta los datos recolectados para que puedan ser transferidos hacia un servidor de aplicaciones;
  - (2) Procesamiento y **almacenamiento de información**, que permita evaluar los datos recolectados, aplicando reglas de contaminación ambiental y almacenarlos en un sistema de gestión de bases de datos relacional; y
  - (3) Representación gráfica, que determine la **localización geográfica** del prototipo a través de un mapa, y que utilice una interfaz gráfica de usuario para visualizar la información almacenada.

Configuraciòn del experimento

- **Diseño del sistema**
  - Se identificaron **cuatro actores** que interactúan tanto con la aplicación Web como con el dispositivo electrónico;
  - El **usuario visitante** puede visitar el portal Web y consultar la información almacenada en la base de datos;
  - El **usuario avanzado** ejecuta las acciones del usuario visitante y además puede registrarse en la aplicación Web y gestionar los sensores del dispositivo electrónico;
  - Los **dos restantes actores** como son la plataforma Arduino y los sensores, son elementos electrónicos que interactúan mediante la API desarrollada en C/C++.

Configuraciòn del experimento



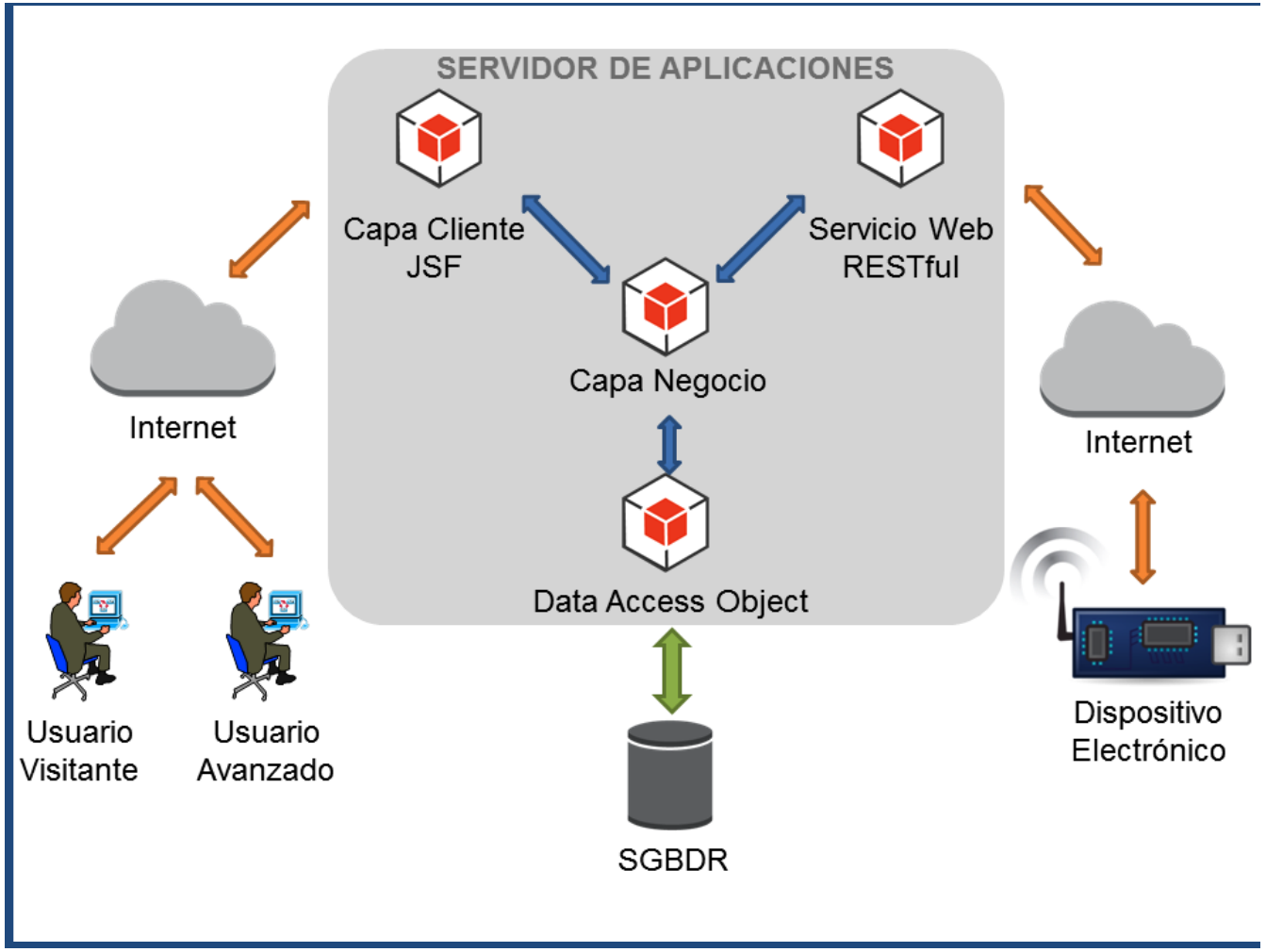
Configuraciòn del experimento

- **Diseño del sistema**
  - El uso de **patrones de diseño** ayuda a lograr un diseño de software reutilizable. En este proyecto se utilizaron los siguientes patrones de diseño:
  - (1) El patrón de diseño **Modelo Vista Controlador** (MVC) se empleó para diseñar las capas de negocios y presentación.
  - (2) El patrón de diseño **Data Access Object** (DAO) que se lo utilizó para acceder a la capa de datos y está separado de la capa de negocios;

Configuraciòn del experimento

- **Arquitectura del software**
  - Es una representación que permite analizar la **efectividad del diseño** para cubrir los requerimientos, efectuar cambios, y reducir los riesgos al momento de construir el software;
  - La elección de utilizar una **arquitectura en capas** y dividir a un sistema en subsistemas, es una estrategia que presenta varias ventajas como la reutilización de componentes, la facilidad en el mantenimiento y la reducción del impacto que podrían generar los cambios en los requerimientos;
  - Utilizando estas premisas, la aplicación ha sido dividida en 4 capas, cada una con funciones específicas y que se comunican entre sí para proporcionar la **interacción** entre el hardware (Arduino) y el software (servidor de Aplicaciones).

Configuraciòn del experimento



Configuraciòn del experimento

- **Pruebas**
  - En la línea base del producto, **Scrum** permitió:
  - (1) definir el producto resultante o prototipo de cada iteración;
  - (2) priorizar y centrar el esfuerzo en las tareas importantes para conseguir una real funcionalidad de cada entregable; y
  - (3) gestionar el tiempo de tal manera que se pueda cumplir con las fechas límite para cada entrega.
  - Otra práctica importante constituyó la aplicación de las pruebas a cada prototipo, Product Increment, para asegurar su funcionamiento y retroalimentación para los ajustes en el diseño.

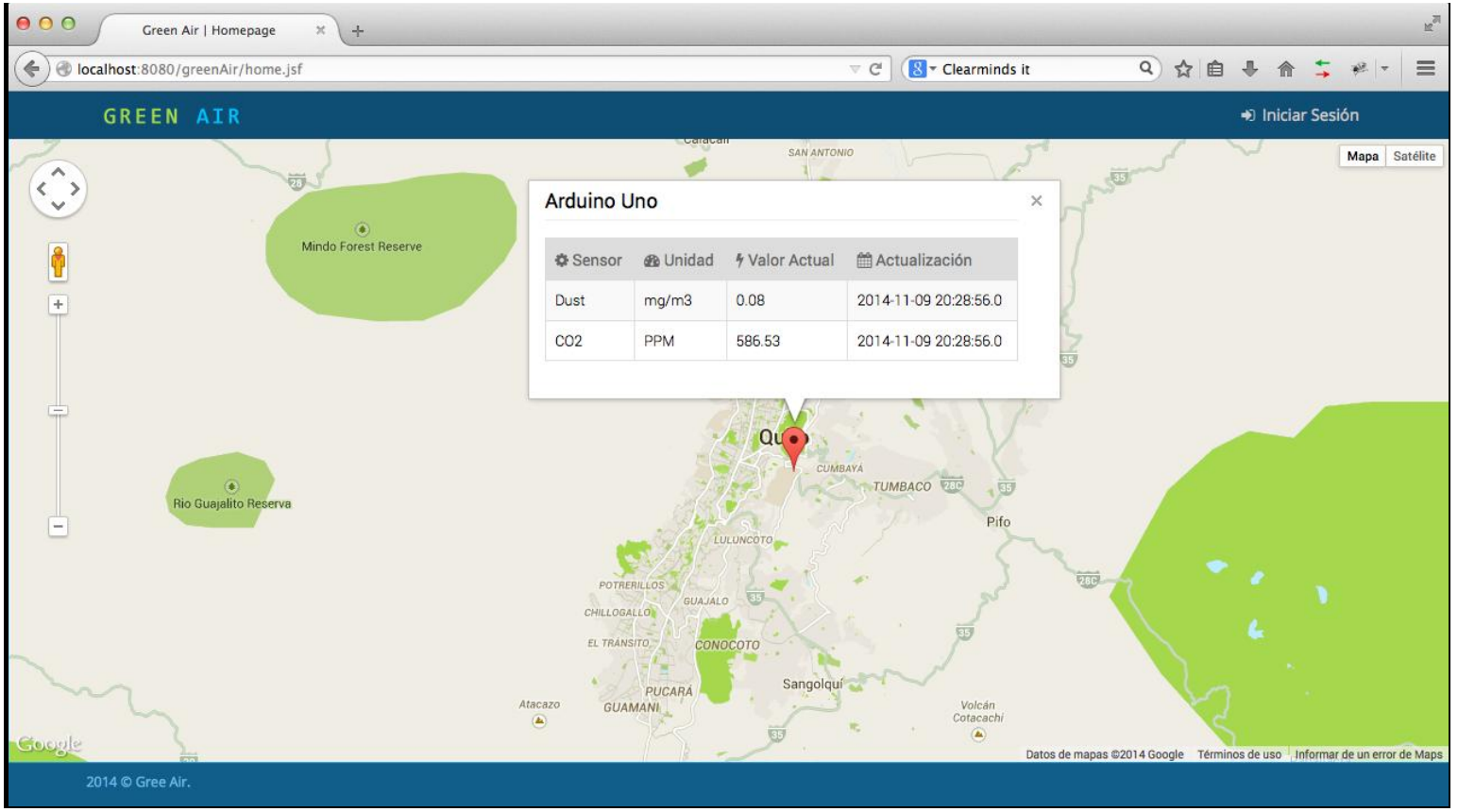
Configuraciòn del experimento

- **Pruebas**
  - En la línea base del producto, **XP** permitió:
  - (1) pruebas unitarias aplicadas a cada uno de los subsistemas, que componen la solución, para evaluar su correcto funcionamiento. Para esta tarea, se emplearon dos herramientas de software compatibles con la plataforma *Java Enterprise Edition (JEE)*: *JUnit* y *Arquillian*;
  - (2) integración continua, que permitieron asegurar la calidad del software que se desarrolló en cada una de las tres iteraciones ejecutadas.

Configuraciòn del experimento

- **Pruebas**
  - En el último Sprint se abordó las dos últimas **historias de usuario** contenidas en la pila de producto o Product Backlog.
  - El gráfico **Burn-Down Chart** se empleó como herramienta de seguimiento para medir el avance del proyecto del Sprint en ejecución.
  - Se cuantificó el número de tareas del Sprint Backlog en relación al tiempo de entrega planificado para determinar el **grado de cumplimiento** del cronograma.

Configuraciòn del experimento



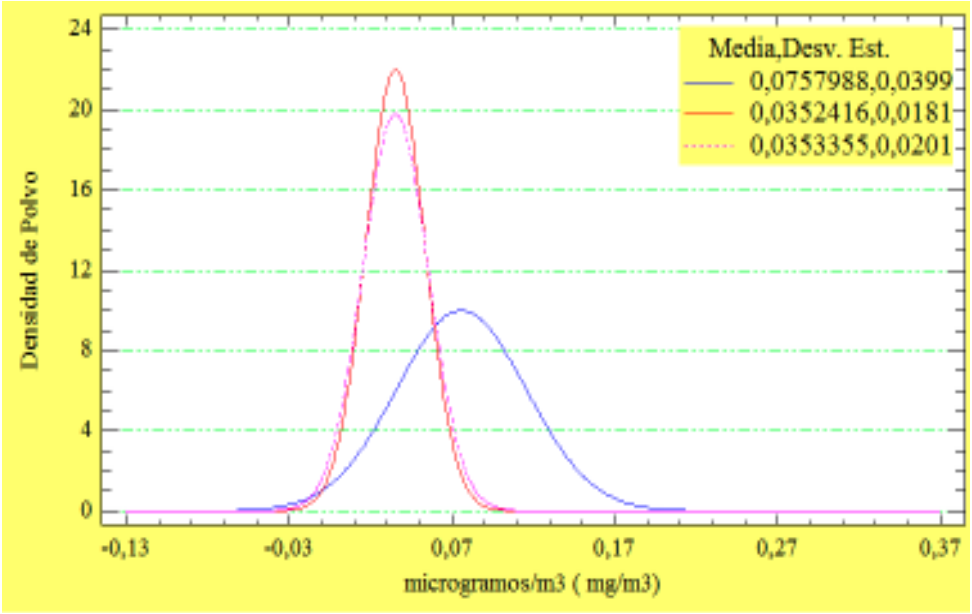
La Figura muestra el demo de la última iteración o Sprint 3.

Evaluaciòn de resultados

- Las **mediciones**, producidos por el sistema de monitoreo inalámbrico, se realizaron en tres lugares diferenciados como la parroquia Jipijapa de la ciudad de Quito, el Castillo de Amaguaña; y las Cuevas de Investigación científica (Gruta de la Virgen) en la ciudad del Tena, provincia del Napo, Ecuador;

- Los resultados fueron comparados con la Normativa de la Organización Mundial de la Salud (OMS), el estándar ASHRAE-62 para el aire en espacios internos; y el estándar de la Agencia de Protección del Medio Ambiente de USA (US-EPA), que mide la **concentración promedio de aire** en espacios externos.

Evaluaciòn de resultados



- Existe una inmensa cantidad de CO_2 en el **ambiente subterráneo**, el cual apenas permite la presencia en el mismo, es decir, con este tipo de mediciones in situ se puede monitorear en tiempo real amenazas por este gas potencialmente letal.

Evaluaciòn de resultados

- En resumen, los resultados muestran que utilizando el sistema de monitoreo inalámbrico en el sector de Jipijapa, la calidad del aire se encuentra bajo los límites permisibles establecidos por las **normas internacionales** publicadas por US-EPA. En el caso de las cuevas, se revela que la concentración de CO_2 supera por largo el valor referencial de la media atmosférica de 400 partes por millón y la densidad de polvo, sin embargo es menor, a la media que se presentó la ciudad de Quito.

- Una importante restricción identificada fue con respecto a la capacidad de procesamiento y almacenamiento con la que cuenta Arduino. lo que justificó la implementación de **servicios Web tipo REST con formatos JSON.**

Conclusiones

- El objetivo de esta investigación fue aplicar un proceso de desarrollo de software para lo cual se utilizaron de forma conjunta las **metodologías ágiles Scrum y XP**, con el objetivo de construir un sistema inalámbrico para el monitoreo de contaminación del aire en tiempo real.
- Con la combinación de ambas metodologías se buscó alcanzar los objetivos planteados en el menor tiempo posible entregando **un producto totalmente funcional** conforme a los requerimientos planteados.
- El prototipo incluyó un sistema informático, basado en una **arquitectura de n-capas**, cuyo hardware y software son de código abierto y de bajo costo.

Conclusiones

- Con este modelo de software y el uso del hardware basado en Arduino, se buscó **abaratar los costos en el mercado** y ofrecer soluciones más económicas que compitan con soluciones comerciales.

- La prueba de concepto se la aplicó en las ciudades de Quito, Amaguaña y Tena del Ecuador, **con resultados positivos**, tanto en el funcionamiento del software y hardware que componen el sistema de monitoreo, como en la medición referencial de la concentración de los contaminantes del aire.

Trabajo Futuro

- Como trabajo futuro se planea implementar software y hardware para una red de nodos inalámbricos, cuyos sensores realicen el monitoreo ambiental, y los datos generados sean almacenados en la **nube computacional**.

- Adicionalmente, se esta ejecutando el proyecto para implementar una **red de nodos inalámbricos**, interconectados entre sí, cuyos sensores realicen el monitoreo del tráfico vehicular, y aplicando técnicas de minería de datos, permita para la predicción del tráfico en la Av. Simón Bolívar de la ciudad de Quito.

Alguna pregunta?

Gracias!