**BotBasic 1.0**

abril, 2017

**¿Qué es BotBasic?**

BotBasic es una plataforma y un lenguaje para el desarrollo rápido aplicaciones móviles basadas en chat.

En la actualidad una vasta mayoría de usuarios utiliza apps de chat cotidianamente. El uso de chatapps es parte del conocimiento que los usuarios de Internet ya poseen.

BotBasic permite desarrollar aplicaciones móviles que se usan por medio de chatapps. Cada aplicación se implementa por medio de "conversaciones" que dirigen la interacción con el usuario. Actualmente Telegram es la chatapp soportada en BotBasic.

**¿Cómo se implementa una aplicación en BotBasic?**

BotBasic provee un lenguaje de programación basado en el conocido lenguaje BASIC, con extensiones que permiten la implementación de aplicaciones conversacionales en chatapps y la conexión con modelos de negocios.

Cada modelo de negocio es realizado, de acuerdo a la solución requerida, a través del diseño e implementación de:

1. Una base de datos relacional, con el lenguaje SQL.
2. Un conjunto de rutinas en PHP, denominadas "primitivas de BotBasic", que permiten acceder y modificar los datos de la base de datos relacional, así como efectuar operaciones asociadas a la lógica de negocios.
3. Un programa elaborado en BotBasic, que refleja la lógica conversacional y a través del cual se accede a las primitivas implementadas en PHP.
4. Opcional: una aplicación web (webapp) de administración, que permite efectuar operaciones que por su naturaleza de manejar datos voluminosos no se adaptan a una interfaz de chat conversacional.

Cada programa en BotBasic se compone de varios subprogramas, uno para cada rol que un usuario de un chatapp puede tener dentro de una solución. Una facilidad que ofrece BotBasic es la posibilidad de que un mismo usuario, para un mismo rol, maneje diferentes canales de chat, a fin de efectuar diferentes operaciones y sobre todo para mantenerles historiales separados de chat dentro de su chatapp (FUNCIONALIDAD AÚN NO PROBADA).

Telegram es una chatapp que provee encriptamiento end-to-end entre usuarios y entre usuarios y el servidor en el que reside la solución de BotBasic. A través de Telegram viajan tanto mensajes de texto así como imágenes, fotos, videos, clips de audio y de voz, documentos de todo tipo, coordenadas de ubicación física y contactos telefónicos. Todos estos contenidos pueden ser manejados por BotBasic (FUNCIONALIDAD AÚN NO PROBADA).

**Arquitectura de BotBasic**

BotBasic provee un componente de plataforma de ejecución de aplicaciones (BotBasic apps) y un lenguaje para el diseño e implementación de las conversaciones y sus modelos de negocios.

Los modelos de negocios son implementados por medio de PHP y bases de datos SQL. Las bases de datos representan la estructura y el contenido de los datos asociados a cada modelo de negocios (el cual es modelado durante el análisis en cualquier proyecto de implementación de software). Los repositorios de bases de datos pueden residir en el mismo servidor físico de BotBasic o en un servidor externo de bases de datos relacionales.

En PHP, la forma de acceder a los datos (creación, lectura, actualización y borrado) son implementados por medio de clases de PHP y se puede utilizar PHP Data Objects (PDO) para ello, a fin de lograr abstracción con respecto a los diferentes manejadores de bases de datos relacionales disponibles en el mercado.

La lógica de negocios es implementada también, en parte, en PHP, por medio de construcciones del lenguaje BotBasic llamadas "primitivas" (incluyendo un tipo de primitivas llamada "menú") y "variables mágicas". Estas construcciones permiten la interacción entre el código elaborado en BotBasic y el modelo de negocios.

Otra parte de la lógica de negocios se implementa directamente en BotBasic. BotBasic es un lenguaje de propósito específico que a la vez es una extensión de un lenguaje de propósito general ampliamente conocido por su facilidad de uso, Basic. La gama de aplicaciones que se puede implementar con BotBasic sólo está restringida por el medio de interacción con el usuario, que son las chatapps.

Toda la "lógica conversacional" se implementa en BotBasic. Si se suma a esto el hecho de que la mayor parte de los cambios en una aplicación tiene que ver con las interfaces de usuario, que en lenguaje de aplicaciones basadas en chat son las "conversaciones", se puede observar que la mayor parte del mantenimiento funcional de una BotBasic app se realiza en lenguaje BotBasic (y no en PHP).

BotBasic, como lenguaje, es compilado y no interpretado. Esto quiere decir que muchos de los problemas de programación de software son detectados previo a la puesta en operación de un código. Un programa hecho en BotBasic sólo es aceptado por el compilador cuando no genera ningún tipo de error. Lo que el compilador de BotBasic verifica es el uso correcto de la "gramática del lenguaje", el ajuste de las cantidades de parámetros de entrada y salida de cada primtiva y los tipos de los argumentos que cada una de las "sentencias" (instrucciones) de BotBasic disponibles puede aceptar. Esta aproximación reduce los problemas y costos asociados a la puesta en producción de nuevas soluciones y actualizaciones a las existentes.

Los programas elaborados en BotBasic se tipean sobre LibreOffice. En cada programa hay secciones correspondientes a valores por defecto de variables (útil para reflejar todos los mensajes que se muestran al usuario, dado que se maneja un sistema de plantillas con componentes extraidos de valores de variables y de retorno de primitivas), primitivas, menús, variables mágicas (cada una es una variable de Basic cuyo proceso de uso -asignación y consulta de valor- está atado a una rutina en PHP que puede interactuar "a la sombra" con el modelo de negocios), y por último el código en sí de cada rol de usuario ("bot" de BB) elaborado según la gramática del lenguaje.

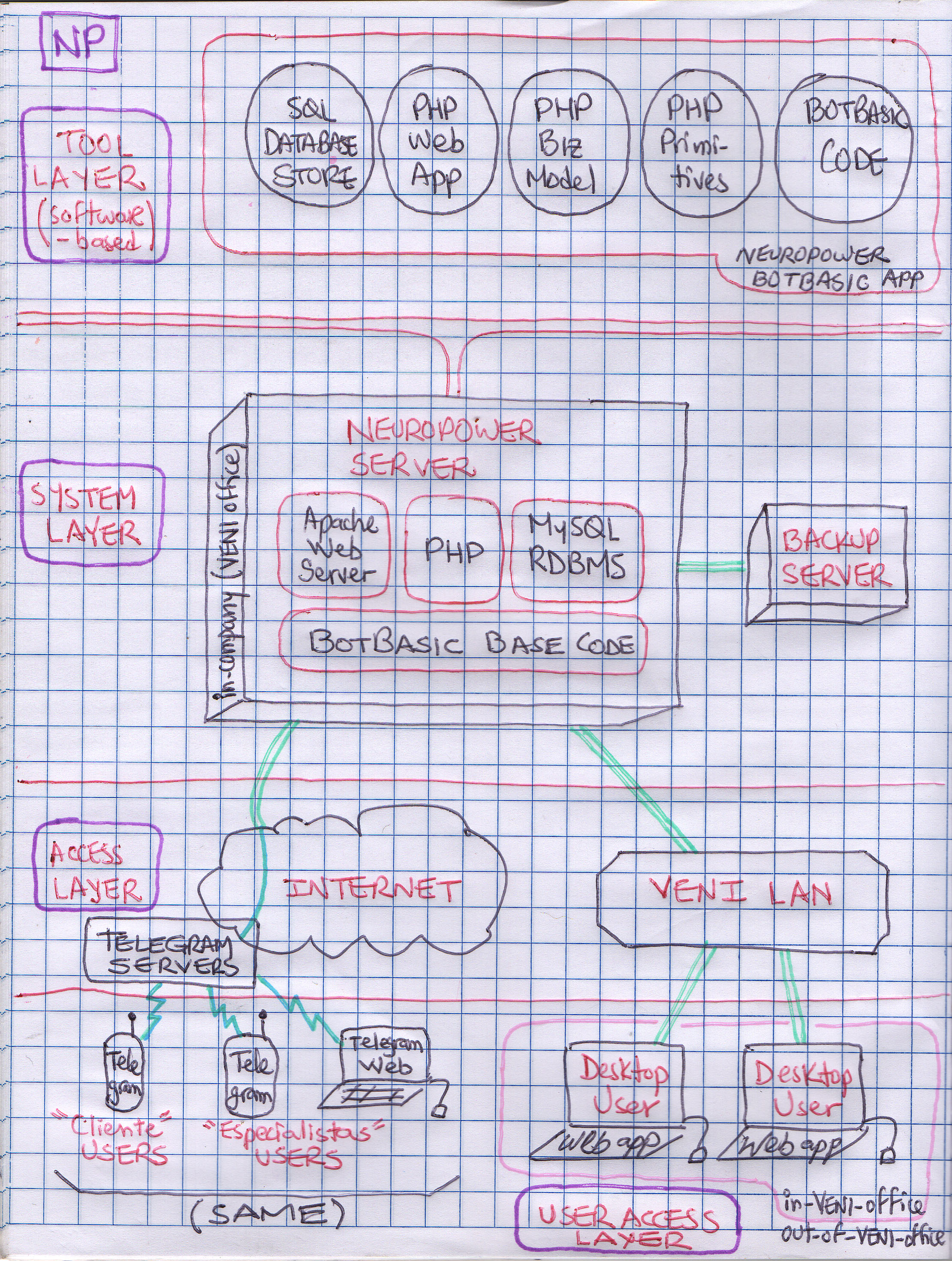
A veces las soluciones requieren back-ends administrativos, que son utilizado por pocos usuarios en forma de aplicaciones web sobre ambientes de escritorio. Estos back-ends típicamente incluyen dashboards, manejadores de usuarios, gestores de tablas maestras del modelo de negocios y visores de eventos, pero no mucho más que ello, ya que toda la operación asociada al modelo de negocio es implementada en BotBasic para ser utilizada a través de chatapps en dispositivos móviles -o sus equivalentes "espejo" web o de escritorio-, con el ocasional y opcional -según el caso- suministro a los usuarios de URL's dinámicos, dentro de las conversaciones, que se abren con un toque en el navegador de Internet del teléfono para mostrar información de carácter voluminoso.

La arquitectura de BotBasic puede ser representada como en el diagrama de la sigueinte página (ejemplo para una BBapp "Neuropower").

Cuatro capas (*layers*) agrupan a los componentes de la solución de manera física y lógica. Las capas están conectadas con enlaces de red. Con doble línea se ha rotulado a la red de las oficinas donde está el servidor de BB y con línea simple a los enlaces de Internet.

El *system layer* incluye a los componentes donde físicamente se ubican los programas y datos del sistema. Los programas (código base de BotBasic y código de la BotBasic app Neuropower en lenguajes PHP, SQL y BotBasic) residen en el servidor principal. Allí también está instalado un sistema operativo Linux y servicios como el servidor web (Apache) y el motor del lenguaje PHP. Dentro del mismo servidor correo un RDBMS Server (software) MySQL.

Desde el *user access layer*, los usuarios se conectan desde sus dispositivos móviles a las BotBasic apps a través de una chatapp, por medio de los enlaces de datos que usan normalmente. Típicamente son redes WiFi o enlaces 3G/4G.



Internet, que es parte del *system access layer*, lleva de ida y vuelta los mensajes a los servidores del proveedor de la chatapp, que los releva hacia el servidor principal de Neuropower. Este servidor tiene habilitada una dirección IP y puerto TCP de entrada.

El servidor también requiere acceder, como cliente, a los servidores del proveedor de las chatapps. Para esta ruta utiliza la conexión saliente hacia Internet de las oficinas donde está el servidor de BB.

Relativo también al *user access layer*, Neuropower podría incluir una Web App de administración, por medio de la cual se gestionan operaciones que requiere un manejo de datos voluminosos y que tienen que ver con monitoreo de eventos, estadísticas agregadas y edición de tablas maestras. El acceso a esta Web App está restringido a los equipos de escritorio de las oficinas mencioandas a través de la LAN.

La capa de aplicación o *tool layer* contiene exclusivamente a Neuropower como BotBasic app. Como toda BotBasic app, está compuesto por cinco componentes:

1. Un SQL Database Store o repositorio de datos.
2. La Web App para las operaciones de administración.
3. El Business Model, implementado como clases en PHP, que aporta lógica de negocio no conversacional y acceso a datos.
4. Las primitivas de BotBasic, implementadas como métodos en PHP, que pueden ser llamadas desde código en lenguaje BotBasic y enlazas con el Business Model.
5. El código elaborado en BotBasic, que representa la lógica conversacional de la aplicación.

Internamente, BotBasic conecta cinco entidades fundamentales:

1. Parser: convierte un archivo de hoja de cálculo que contiene un programa en BB, en una representación interna que permite a las otras entidades funcionar. El parser se ejecuta sólo cuando hay cambios en una versión de un programa en BB. El resto de las entidades son ejecutadas cada vez que un usuario interactúa con una BBapp a través de una chatapp.
2. BotBasicRuntime (o "runtime"): el núcleo de funcionamiento de BB se concentra aquí. Condensa la mayor parte del "estado de ejecución" que está asociado a un usuario de la BBapp desde el punto de vista del modelo de negocios (bizModelUser).
3. BotBasicChannel (o "bbc"): un runtime, y un bizModelUser, tiene al menos un bbc. Puede tener varios para los casos en los que un bizModelUser maneja varios canales de chateo en forma simultánea con la BBapp. Aquí también se guarda parte del estado de ejecución de la BBapp; en particular, el que está asociado al bbc.
4. ChatMediumChannel (o "cmc"): cada bbc se expresa como un "hilo" de chat en una chatapp.
5. ChatMedium (o "cm"): una chatpp como Telegram o uno de los simuladores desarrollados para depurar el código de BB.

Si hay "m" cantidad de ChatMedia y "N" cantidad de BotBasicChannels, entonces habrá m \* N cantidad de ChatMediumChannels.

