

Taller construccion de prototipo

Jorge Armando Toloza Carvajal

Análisis y desarrollo de software

Ficha # 2627106

Instructor: Carlos Ernesto Lizarazo Sierra

SENA

Regional Santa fe de Antioquia

Junio 2023

1. ¿Cuál es el concepto de usabilidad en software?

La usabilidad en software se refiere a la medida en que un producto de software puede ser utilizado por los usuarios de manera efectiva, eficiente y satisfactoria. Se enfoca en la facilidad de uso, la capacidad de aprendizaje, la eficiencia en la ejecución de tareas, la satisfacción del usuario y otros aspectos relacionados con la experiencia del usuario al interactuar con el software. Un software con alta usabilidad es intuitivo, fácil de aprender, eficiente en el uso y proporciona una experiencia positiva al usuario.

2. ¿Cuáles son los estándares de calidad de software?

Existen varios estándares de calidad de software reconocidos a nivel internacional, algunos de los cuales incluyen:

ISO 25010: Esta norma establece un conjunto de características y subcaracterísticas para evaluar la calidad del software, que abarcan aspectos como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

CMMI (Capability Maturity Model Integration): Es un modelo de mejora de procesos que evalúa la capacidad y madurez de las organizaciones de desarrollo de software. Proporciona una estructura para mejorar los procesos y prácticas de desarrollo, gestión y soporte.

IEEE 730: Esta es una norma que define el estándar de calidad para la documentación de software. Proporciona pautas y criterios para la creación de documentos de calidad que describen los requisitos, el diseño, las pruebas y otros aspectos del software.

Six Sigma: Aunque originalmente desarrollado para la mejora de procesos en la fabricación, Six Sigma se ha aplicado al desarrollo de software para mejorar la calidad mediante la reducción de defectos y la mejora de la eficiencia.

3. ¿Qué tecnologías existen para implementar calidad?

Existen varias tecnologías y enfoques utilizados para implementar y garantizar la calidad en el desarrollo de software. Algunas de estas tecnologías incluyen:

Pruebas automatizadas: Se utilizan herramientas y marcos de prueba automatizados para realizar pruebas funcionales, de rendimiento, de carga y otros tipos de pruebas de manera eficiente y repetible. Esto ayuda a identificar y corregir errores y garantizar un alto nivel de calidad.
Control de versiones y gestión de configuración: El uso de sistemas de control de versiones, como Git, y herramientas de gestión de configuración ayuda a mantener un registro de los cambios realizados en el software y permite un seguimiento eficiente de las versiones y configuraciones utilizadas.

Integración continua y entrega continua: Estas prácticas implican la automatización de la construcción, prueba y despliegue de software de manera continua. Ayudan a detectar problemas de calidad de manera temprana y aseguran la entrega regular y confiable del software.

Análisis estático y revisión de código: Las herramientas de análisis estático de código ayudan a identificar problemas de calidad y buenas prácticas en el código fuente sin necesidad de ejecutar el software. La revisión de código por parte de pares también es una práctica común para identificar errores y mejorar la calidad del software.
Monitoreo y análisis de rendimiento: Se utilizan herramientas y técnicas de monitoreo para evaluar el rendimiento del software en tiempo real, identificar cuellos de botella

Reglas de Usabilidad y Accesibilidad para el Software de Call Center:

4. **Regla de Consistencia:** Consiste en mantener una interfaz coherente en todo el software para facilitar la navegación y la comprensión. Esto implica utilizar una estructura y diseño visual consistentes, así como una terminología uniforme en todo el sistema.

5. **Regla de Simplicidad:** Busca simplificar la interfaz y las funcionalidades del software para facilitar su uso. Evitar la sobrecarga de información, minimizar los pasos necesarios para realizar tareas y presentar la información de manera clara y concisa.

6. **Regla de Retroalimentación:** Proporcionar una respuesta visual o auditiva inmediata al usuario después de realizar una acción. Esto puede incluir mensajes de confirmación, indicadores de progreso, notificaciones de error y cualquier otra forma de comunicación para mantener al usuario informado sobre el estado de sus acciones.

7. **Regla de Flexibilidad:** Permitir al usuario personalizar y adaptar la interfaz según sus preferencias y necesidades individuales. Proporcionar opciones de configuración, como ajustes de tamaño de fuente, contraste de colores y preferencias de accesibilidad, para asegurar que el software sea utilizable por una amplia gama de usuarios.

8. **Regla de Eficiencia:** Buscar minimizar el tiempo y los esfuerzos requeridos por el usuario para realizar tareas en el software. Utilizar atajos de teclado, proporcionar funcionalidades de búsqueda y permitir la reutilización de datos y configuraciones para agilizar las actividades del usuario.

9. **Regla de Legibilidad:** Garantizar que el contenido del software sea fácilmente legible y comprensible. Utilizar un tamaño de fuente adecuado, un contraste de colores adecuado y una tipografía clara. También se pueden proporcionar opciones de cambio de fuente y tamaño para adaptarse a las preferencias de los usuarios.

10. Diferencia entre una aplicación independiente y una aplicación web:

Una aplicación independiente, también conocida como aplicación de escritorio o de cliente, es un software instalado y ejecutado en el dispositivo del usuario. La aplicación se desarrolla específicamente para un sistema operativo particular y generalmente no requiere una conexión a Internet para funcionar. Las aplicaciones independientes suelen ser más rápidas y tienen un acceso directo a los recursos del dispositivo, pero pueden requerir actualizaciones periódicas para mantenerse al día.

Por otro lado, una aplicación web es un software que se ejecuta en un navegador web. Estas aplicaciones están alojadas en un servidor y se accede a ellas a través de Internet. Las aplicaciones web son multiplataforma, lo que significa que se pueden utilizar en diferentes dispositivos y sistemas operativos sin necesidad de instalar software adicional. Sin embargo, su rendimiento y acceso a los recursos del dispositivo dependen de la conexión a Internet y el navegador utilizado.

Características de cada tipo de aplicación:

a. Características de una aplicación independiente:

- ✓ Se instala y se ejecuta en el dispositivo del usuario.
- ✓ Tiene acceso directo a los recursos del dispositivo, como la memoria y el almacenamiento.

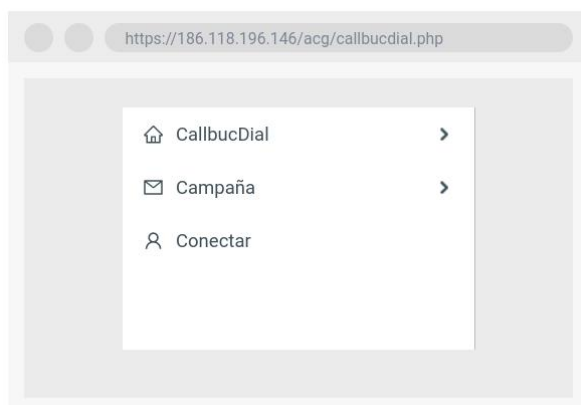
- ✓ Puede funcionar sin conexión a Internet, en la mayoría de los casos.
- ✓ Requiere actualizaciones periódicas para mantenerse actualizada.
- ✓ Proporciona una experiencia más rápida y fluida.

b. Características de una aplicación web:

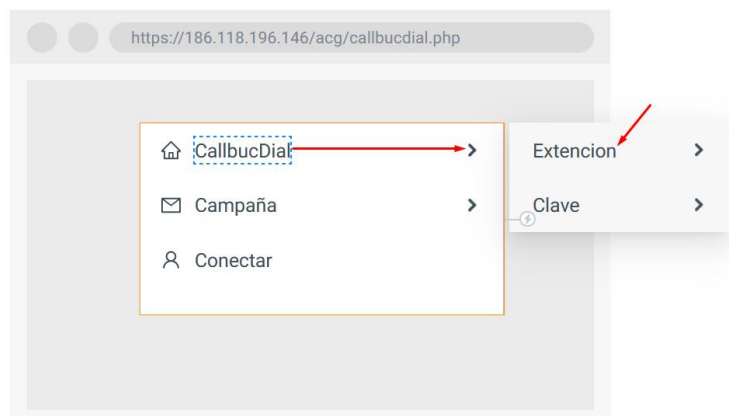
- ✓ Se accede a través de un navegador web en cualquier dispositivo con conexión a Internet.
- ✓ No requiere instalación en el dispositivo del usuario.
- ✓ Puede ser utilizada en diferentes sistemas operativos y dispositivos.
- ✓ No tiene acceso directo a los recursos del dispositivo y depende de las capacidades del navegador.
- ✓ No necesita actualizaciones del usuario, ya que las actualizaciones se realizan en el servidor.

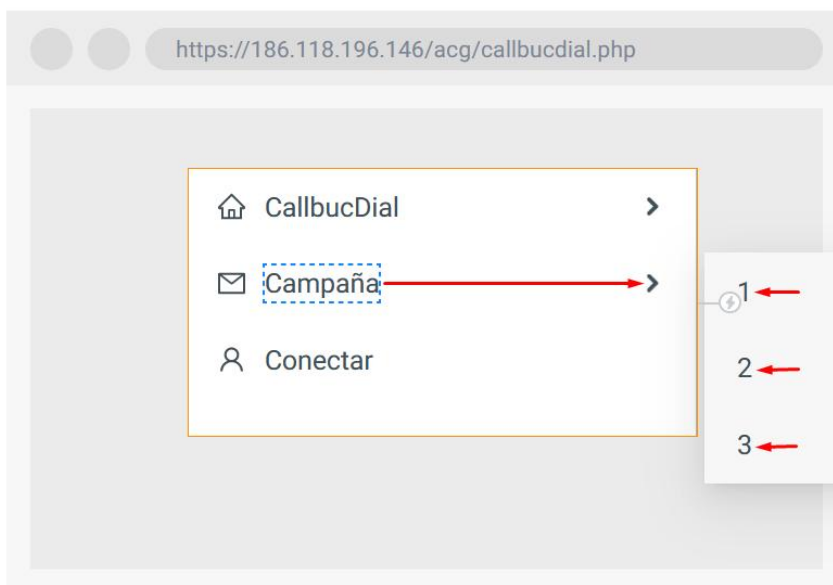
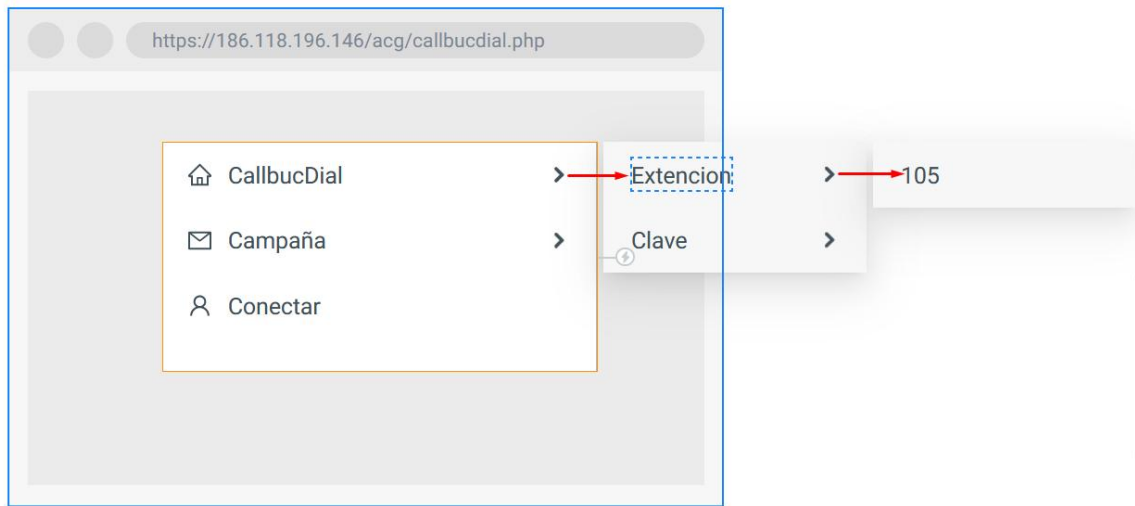
Es importante tener en cuenta que tanto las aplicaciones independientes como las aplicaciones web pueden cumplir con los principios de usabilidad y accesibilidad, pero cada una tiene sus propias características y consideraciones técnicas. La elección entre una aplicación independiente y una aplicación web depende de los requisitos específicos del proyecto, las necesidades de los usuarios y las limitaciones técnicas.

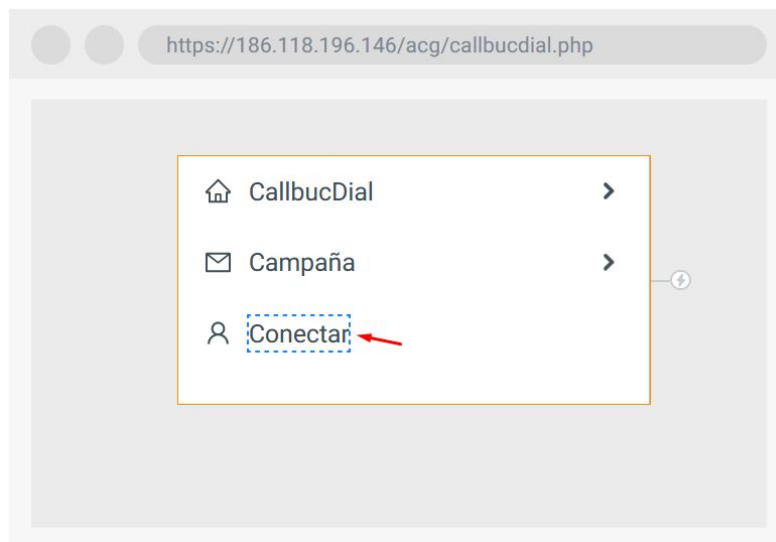
11. Presentacion en pantalla atraves del prototipo de software diseñado en MOCKITT



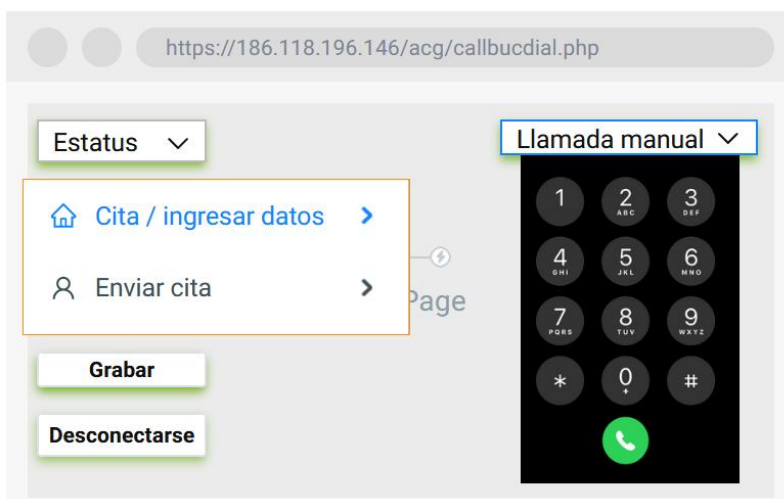
11.1 Presentacion en pantalla de inicio de sección y conexión a campaña según el prototipo MOCKITT



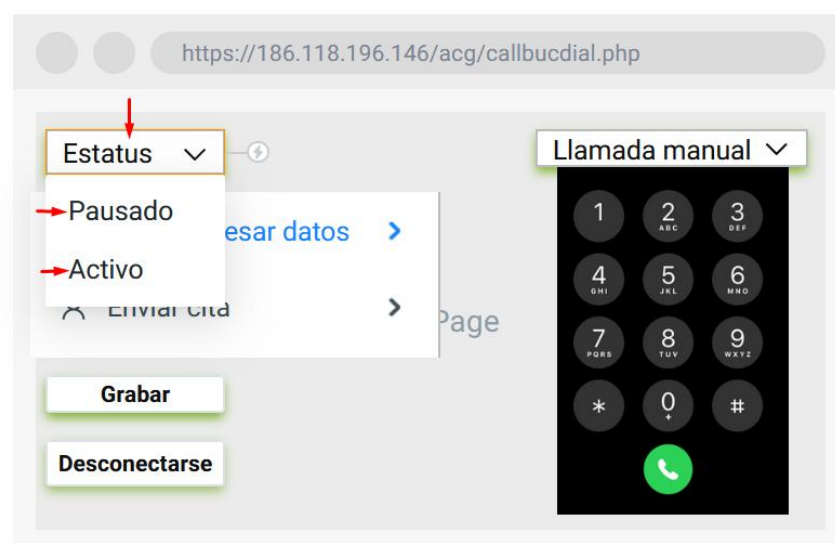




11.2 Presentacion en pantalla una vez el asesor esta conectado según el prototipo MOCKITT



11.3 Presentacion en pantalla para pasar de **Pausado** por defecto a **activo** de modo que el predictivo le pueda enviar llamadas

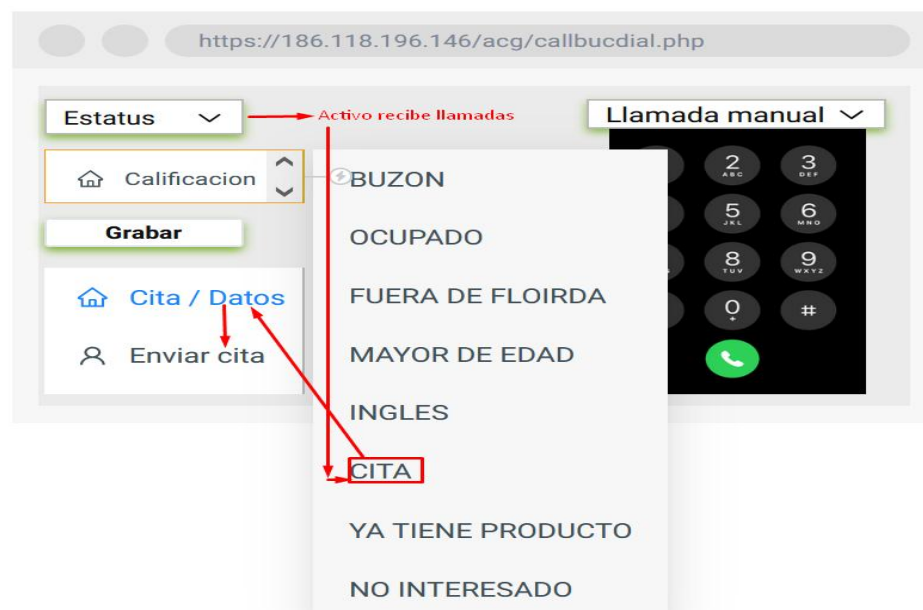




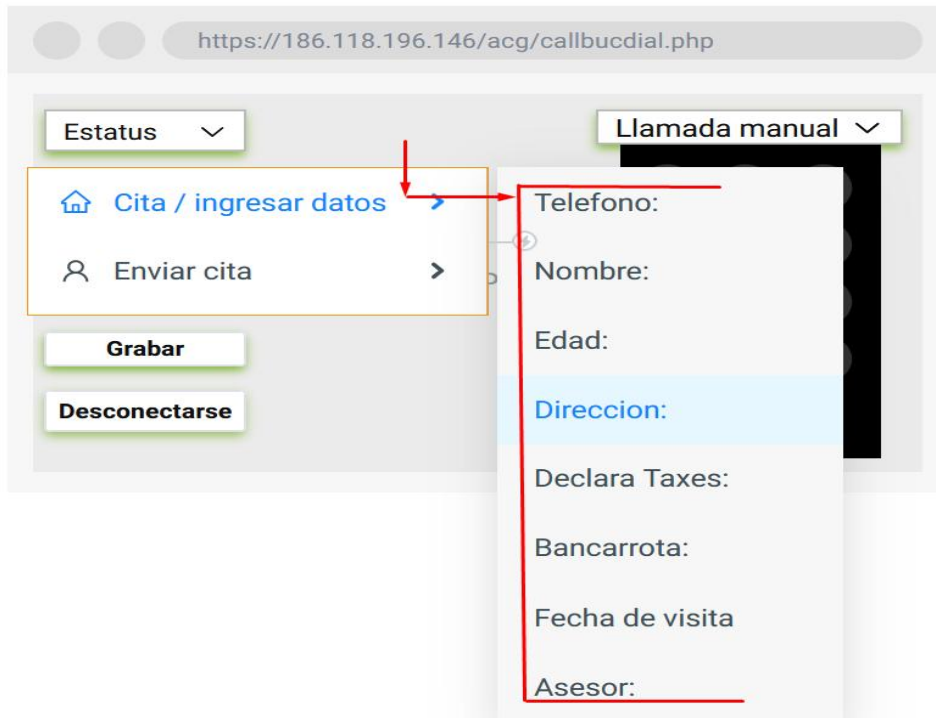
11.4 Presentacion en pantalla para indicar que cuando el asesor esta Pausado puede hacer llamadas manuales o tomar un descanso



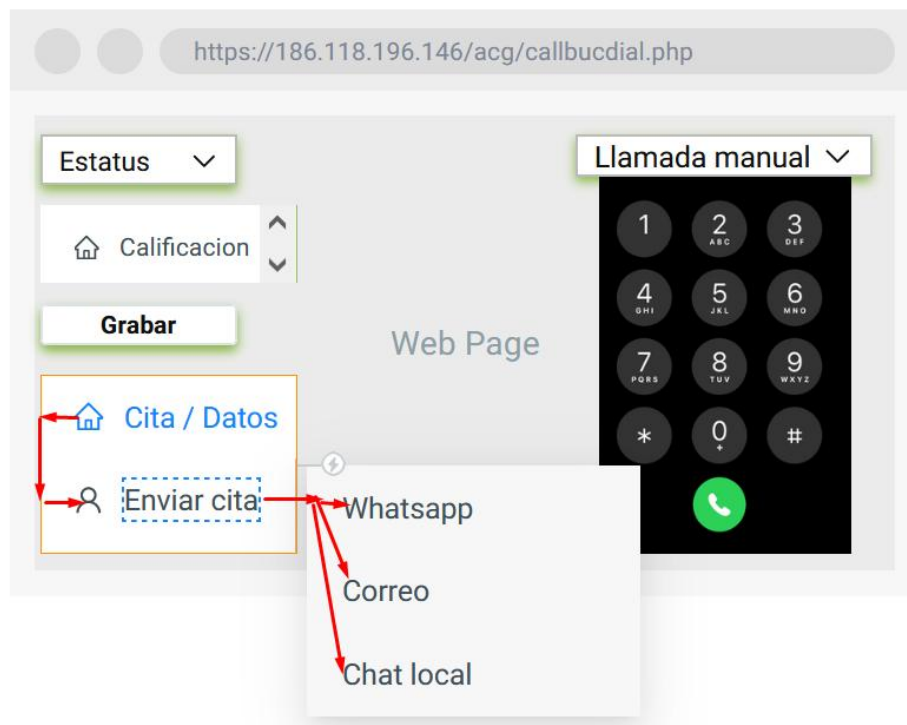
11.5 Presentacion en pantalla para mostrar que cada llamada que el asesor reciba podrá ser calificada según corresponda



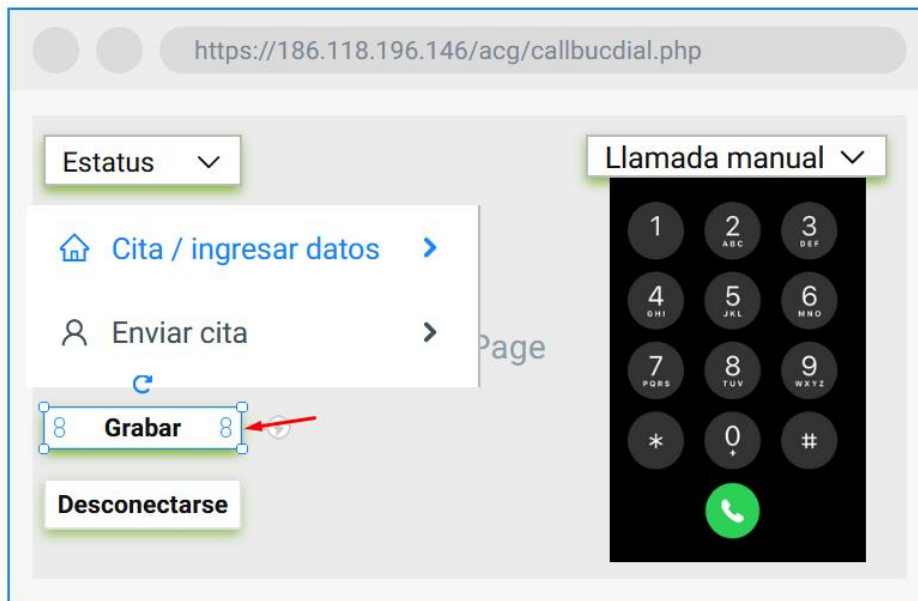
11.6 Presentacion en pantalla para el caso en el que el asesor logre convencer al cliente, deberá dar clic en la opción cita y esto le redirigirá a la opción datos de cita



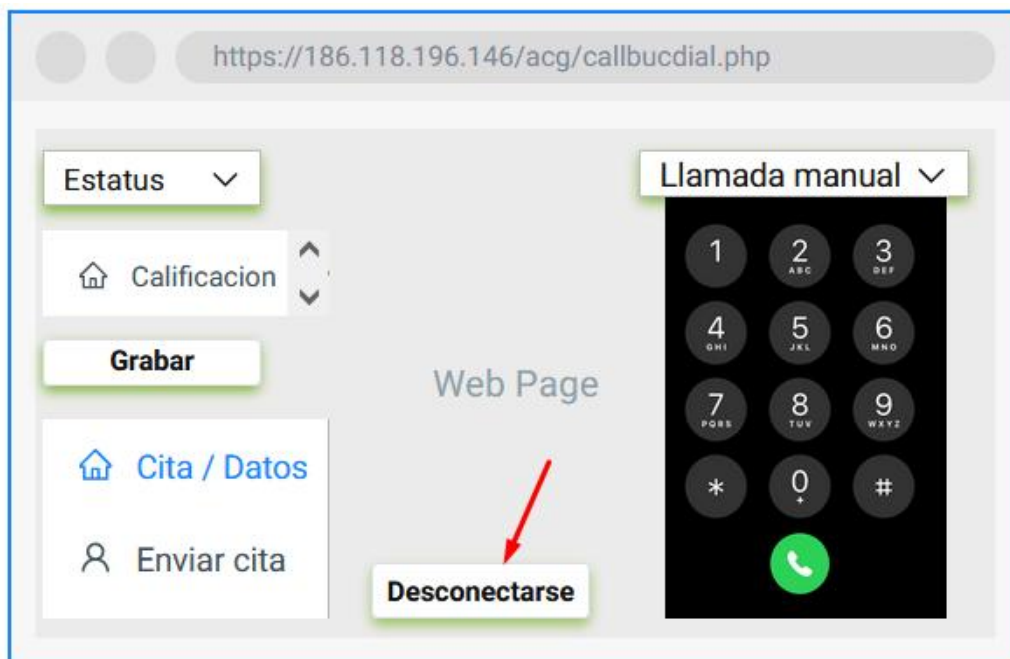
11.7 Presentacion en pantalla de la opción enviar cita, la cual sera útil cuando el asesor concrete una venta o cita, en ese momento tomara los datos y dará clic en la opción enviar cita eligiendo la que el jefe inmediato aya dispuesto



11.8 Presentacion en pantalla de la opción grabar, pára que cada vez que el asesor note que el cliente muestra interés en el producto pueda grabar dicha llamada, de igual modo si el cliente lo desea esta opción se podrá dejar en automático.



11.9 Presentacion en pantalla de la opción desconectarse para el momento que finaliza el turno o el supervisor ordena la desconexión



12. Modelo entidad, relación

