

**Instituto de Computación
Facultad de Ingeniería
Universidad de la República**

DESARROLLO DE NUEVAS INTERFACES PARA MEJORAR LA ACCESIBILIDAD.

Informe de Proyecto de Grado

Montevideo - Uruguay

Jorge Fernández Otero
jorge.fernandez@seciu.edu.uy

Tutores

Ana Martín Salguero
anamartin@fing.edu.uy

Sebastián Marichal Baraibar
sebastian.marichal@seciu.edu.uy

Agosto 2015

Agradecimientos

Quiero expresar mi profundo agradecimiento a las siguientes personas sin las cuales me hubiera sido imposible realizar este proyecto:

A mi familia por el apoyo incondicional, la confianza y el aliento en los momentos de dificultad, la comprensión y paciencia a lo largo de todo el proyecto.

A mis tutores Ana Martín y Sebastián Marichal por su tiempo, su ayuda y orientación en todo momento y salvando distancias.

A la directora de Delia Urgoiti, a todas las maestras y personal que trabaja en la Escuela N° 200, por su colaboración y calidez.

A todos los niños que concurren a la Escuela N° 200 hayan participado o no de las actividades realizadas durante el proyecto, por hacerme recordar lo lindo que es ser niño.

Finalmente, a mis amigos y compañeros de trabajo por el apoyo y la paciencia.

Resumen

El objetivo de este proyecto es desarrollar herramientas libres que permitan mejorar la accesibilidad de algunos dispositivos tecnológicos para que niños y niñas con discapacidad motriz puedan hacer uso de ellos. En particular, el foco de atención será depositado en las computadoras y tablets entregadas por el plan Ceibal con el fin de reducir la brecha en la inclusión tecnológica e intentar mejorar la calidad de vida de estos niños.

El prototipo fue realizado en base a un diseño centrado en el usuario creado de manera iterativa en colaboración con la comunidad de maestras y niños de la Escuela N° 200 Dr. Ricardo Caritat, única escuela pública para niños y niñas con parálisis cerebral. Está implementando en Python utilizando una librería open source multiplataforma llamada Kivy orientada al desarrollo de aplicaciones que hacen uso de interfaces de usuario innovadoras, como por ejemplo las multitáctiles.

Índice de contenido

Capítulo 1.....	8
Introducción.....	8
1.1 Objetivos.....	9
1.2 Aportes del proyecto.....	9
Capítulo 2.....	12
Estado del Arte.....	12
2.1 Accesibilidad educativa de niños y niñas con discapacidad.....	12
2.2 Discapacidad Motriz: Parálisis Cerebral.....	12
2.2.1 Definición y Diagnóstico.....	12
2.2.2 Clasificación.....	13
2.2.2.1 Clasificación fisiopatológica.....	13
2.2.2.2 Clasificación topográfica.....	14
2.2.2.3 Clasificación según grado de limitación funcional.....	14
2.2.3 Incidencia.....	14
2.2.3.1 Parálisis Cerebral Espástica.....	14
2.2.3.2 Parálisis Cerebral Discinética.....	15
2.2.3.3 Parálisis Cerebral Atáxica.....	15
2.2.4 Dificultades asociadas.....	16
2.2.5 Tratamientos y Educación en niños y niñas con P.C.....	17
2.2.6 Pronóstico.....	17
2.3. Diseño Universal y la Educación.....	18
2.3.1 Paradigma del Diseño Universal.....	18
2.3.1.1 Los 7 principios del Diseño Universal [14].....	19
2.3.2 Diseño universal para la Educación.....	21
2.4 Interacción Persona Computadora.....	22
2.4.1 Estilos de Interacción.....	23
2.4.1.1 Línea de comandos.....	23
2.4.1.2 Formularios.....	24
2.4.1.3 Menús y navegación.....	24
2.4.1.4 Manipulación directa.....	25
2.4.1.5 Interacción Asistida.....	26
2.4.2 Paradigmas de Interacción.....	27
2.4.2.1 Computadora Personal.....	28
2.4.2.2 Realidad Virtual.....	28
2.4.2.3 Computación Ubicua.....	28
2.4.2.4 Realidad Aumentada.....	29
2.5 Diseño de Sistemas Interactivos.....	31
2.5.1 Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad (MPIu+a).....	31
2.5.1.1 Análisis de Requisitos.....	32
2.5.1.2 Diseño.....	32
2.5.1.3 Prototipado.....	33
2.5.1.4 Evaluación.....	33
2.5.1.5 Implementación.....	34
2.5.1.6 Lanzamiento.....	34
2.5.2 Diseño orientado a niños.....	34
2.5.2.1 Relevamiento de Requerimientos.....	35

2.5.2.2 Diseño de Interacción.....	35
2.5.2.3 Evaluación.....	36
2.5.3 Diseño orientado a niños y niños con parálisis cerebral.....	36
2.5.4 Desarrollos realizados con un diseño orientado a niños con Parálisis Cerebral.....	37
2.6 OLPC - Plan Ceibal y la Accesibilidad.....	45
2.6.1 Equipos.....	47
2.6.2 Actividades.....	49
2.6.2.1 Hablar/Hablar con Alicia.....	49
2.6.2.2 Escribir Especial.....	49
2.6.2.3 PUI: Perceptual User Interface.....	50
Capítulo 3.....	51
Contexto del Proyecto.....	51
3.1 Patologías de los niños.....	52
3.1.1 Características del aprendizaje.....	52
3.1.2 Comunicación.....	53
3.1.3 Recursos Didácticos y Consideraciones Generales.....	55
Capítulo 4.....	56
Problemática detectada y Descripción de la Solución.....	56
4.1 Proceso de Análisis y Diseño.....	56
4.1.1 Actividad 1: Análisis de Requisitos.....	56
4.1.1.1 Análisis Etnográfico.....	56
4.1.1.2 Plataforma.....	57
4.1.1.3 Objetivos de Usabilidad.....	58
4.1.1.4 Descripción de Requerimientos.....	58
4.1.1.4.1 Requerimientos Funcionales.....	58
4.1.1.4.2 Requerimientos no Funcionales.....	59
4.1.2 Actividad 2: Diseño.....	59
4.1.2.1 Funcionalidades.....	59
4.1.2.2 Diseño Detallado.....	60
4.1.2.2.1 Pantallas.....	61
4.1.2.2.2 Navegación.....	70
4.1.3 Actividad 3: Prototipado y Evaluación con los usuarios del diseño de GUI.....	70
4.1.4 Actividad 4: Análisis.....	71
4.1.4.1 Nueva Funcionalidad: Cambio Automático de Modo.....	71
4.1.4.2 Modelo de Dominio.....	71
4.1.5 Actividad 5: Elección de herramientas de Software.....	72
4.1.5.1 Implementación de Interfaz de Usuario.....	72
4.1.5.1.1 Candidatos.....	72
4.1.5.1.2 Selección de Toolkit.....	74
4.1.5.2 Implementación de la Base de Datos.....	74
4.1.5.3 Implementación de la interacción con otras aplicaciones.....	75
4.1.6 Actividad 6: Arquitectura y Diseño.....	75
4.1.6.1 Arquitectura.....	75
4.1.6.2 Diagrama de Clases.....	76
4.1.7 Fin de Iteración del Proceso de Análisis y Diseño.....	77
Capítulo 5.....	78
Implementación.....	78
5.1 Nombre e Icono de la Aplicación.....	78
5.2 Lenguaje y Plataforma utilizados.....	78
5.2.1 Estructura del código.....	79

5.2.2 Kivy.....	80
5.3 Algoritmos.....	82
5.3.1 Definición del Barrido.....	82
5.3.2 Velocidad del Barrido.....	84
5.4 Envío de Eventos.....	84
5.5 Empaquetando para Android.....	84
5.6 Repositorio.....	85
Capítulo 6.....	86
Test de Usabilidad.....	86
6.1 Objetivos.....	86
6.2 Test.....	86
6.2.1 Ubicación del Test.....	86
6.2.2 Testers.....	86
6.2.2.1 Niños.....	86
Clasificación.....	86
6.2.2.2 Maestras.....	87
6.2.3 Materiales.....	87
6.2.4 Técnicas de Registro.....	87
6.2.5 Protocolo.....	87
6.2.5.1 Líneas generales.....	87
6.2.5.2 Actividades.....	87
6.2.6 Descripción del protocolo.....	88
6.2.6.1 Presentación de la actividad.....	88
6.2.6.2 Finalización de la actividad.....	90
6.7 Resultados.....	90
Capítulo 7.....	93
Conclusiones y Trabajo a Futuro.....	93
7.1 Conclusiones.....	93
7.2 Trabajo a Futuro.....	95
8 Referencias.....	96
Anexos: Anexo I.....	100
Accesibilidad educativa de niños y niñas con discapacidad: Leyes y normativas.....	100
1 Panorama Internacional [62].....	100
1.1 Convención sobre los Derechos de las Personas con Discapacidad (2006).....	100
1.2 Programa del Decenio de las Américas 2006 – 2016: Por los derechos y la dignidad de las personas con discapacidad.....	100
1.3 Estrategia del Decenio de las Unión Europea 2010 – 2020.....	102
2 Panorama Latinoamericano [62].....	103
3 Panorama Nacional [63] [64].....	105
Anexo II.....	107
Registro de Test de Usabilidad.....	107
comunicaME.....	107
Datos Tester.....	107
Preguntas.....	107

Capítulo 1

Introducción

El desarrollo de la tecnología, en particular las tecnologías de la información y la comunicación o TIC, ha generado desde sus comienzos un profundo impacto en la vida cotidiana de las personas. En especial, en el ámbito educativo el uso de estas tecnologías puede contribuir al acceso universal a la educación, la igualdad en la instrucción, el ejercicio de la enseñanza y el aprendizaje de calidad y el desarrollo profesional de los docentes, así como a la gestión, dirección y administración más eficientes del sistema educativo [1].

La brecha digital es la distancia en el acceso, uso y apropiación de las TIC a nivel geográfico, nivel socioeconómico y de género [2]. Esta brecha se traduce en desigualdad, exclusión y división de comunidades, afectando mayormente a las que ya sufren de desigualdad y exclusión, profundizando aún más su problemática. En el ámbito educativo, parece fundamental lograr que esta brecha sea cada vez menor para todas las comunidades niños y niñas para poder brindarles un futuro más equitativo e inclusivo.

El Plan Ceibal [3] iniciado en Uruguay en el 2006 está inspirado en el proyecto *One Laptop Per Child*, conocido por su sigla OLPC [4], que en español significa una computadora por niño, tiene como objetivo promover la inclusión digital con el fin de disminuir esta brecha digital facilitando el acceso a la educación y la cultura. De este modo se intenta promover una mayor justicia social generando condiciones de igualdad en el acceso a la información y herramientas de comunicación, a través de la entrega de una computadora a cada niño y niña en edad escolar y para cada maestro de la escuela pública. Asimismo, la capacitación de los docentes en su uso busca fomentar la elaboración de propuestas educativas que aprovechen el recurso.

Se pretende la integración de la tecnología en el aula, en el centro escolar y en el núcleo familiar, desarrollando una cultura colaborativa entre los niños, los maestros, las escuelas y las familias.

Según el informe anual 2012 de Plan Ceibal, se entregaron 223.578 computadoras con un total de 566.552 beneficiarios, capacitando a 14.000 docentes [5]. Dentro de la población objetivo del plan, se requiere dar un especial énfasis a los niños y niñas con discapacidad con el fin de generar las mismas oportunidades de acceso para ellos. Según datos del censo 2011 en Uruguay hay unos 35.000 niños y niñas menores de 15 años con discapacidad, de los cuales 5.200 tienen discapacidad auditiva o sordera total, 15.000 presentan discapacidad visual o ceguera total, 4.700 tienen discapacidades físicas y 18.000 manifiestan discapacidad intelectual [6].

La efectividad del Plan como herramienta de apoyo a la educación, puede verse fuertemente limitada para estos niños al no contar en sus computadoras con herramientas adecuada a sus habilidades preservadas.

La tecnología asistiva es un término que se utiliza para englobar a todo elemento tecnológico que sea usado para mejorar o mantener las capacidades de una persona con discapacidad. El uso de tecnología asistiva en niños y niñas con discapacidad puede ser fundamental en su desarrollo cognitivo, ya que propone una forma alternativa de interacción con el medio que permite simular o crear nuevas experiencias, formas de comunicar, de aprender y de enseñar [7]. Si bien ya existe

tecnología asistiva que intenta resolver algunos de los problemas [8] con los que cuentan estos niños y niñas, no son del todo adecuadas para las computadoras entregadas por Ceibal. El costo puede ser muy elevado e impide que una familia de clase media o baja, más aún en zonas rurales, acceda a este tipo de herramientas si no es a través de donaciones.

Dotar con herramientas de tecnología asistiva a las computadoras y tablets de Ceibal, dando acceso a los niños y niñas con discapacidad a sus funcionalidades, puede producir una gran transformación habilitadora en esta población dando la posibilidad a un desarrollo personal que tal vez no sería posible de otra manera, utilizando un recurso con el que ya cuentan.

También se desarrollará un prototipo con el fin de validar el diseño de forma experimental.

1.1 Objetivos

El objetivo general del proyecto consiste en diseñar una herramienta siguiendo un diseño centrado en el usuario que mejore la accesibilidad de las computadoras y tablets entregadas por Plan Ceibal para niños y niñas con discapacidad motriz. Se pretende con ello favorecer su inclusión dentro del Plan, aprovechando así el recurso tecnológico como instrumento de su aprendizaje.

Los objetivos específicos son:

- Revisar el estado del arte de las herramientas disponibles para la educación que son utilizados con niños con discapacidad motriz. Esta exploración se hará en particular en aquellas que son de software libre.
- Diseñar una herramienta en conjunto con la comunidad de la escuela especial N° 200 Dr. Ricardo Caritat. De este modo se accede a la información necesaria para guiar el proceso de diseño, como las necesidades de los usuarios, las habilidades y limitaciones en el contexto escolar, el tipo de interacción con el entorno y en particular la forma de trabajar con las computadoras.
- Desarrollar un prototipo que permita testear la usabilidad del producto y que sirva de base para un desarrollo posterior de la herramienta, siendo creado con código abierto, una arquitectura modularizada y que sea fácilmente extensible y mantenible para que sea posible continuar con su desarrollo.
- Definir un protocolo para realizar pruebas de usabilidad sobre el prototipo desarrollado.

1.2 Aportes del proyecto

Los aportes principales de este proyecto son los siguientes:

- Un relevamiento del estado del arte que incluye:
 - Herramientas tanto de hardware como de software para la discapacidad motriz infantil.
 - Situación actual de la discapacidad infantil en lo que respecta a leyes y regulaciones tanto internacionales como nacionales.
 - Comprensión de la discapacidad motriz, sobre todo la causada por la parálisis cerebral la

cual es la de más incidencia en niños y niñas.

- Un diseño de una herramienta centrado en el usuario que mejora la usabilidad de computadoras y tablets de niños con discapacidad motriz.
- Un prototipo desarrollado en código abierto, fácilmente mantenible y extensible.
- Un protocolo específico para realizar pruebas de usabilidad que indican las debilidades de la solución y aportan elementos para un mejor diseño a futuro.
- La propia documentación generada del diseño y el desarrollo, junto con la experiencia obtenida, sirven como antecedente para continuar la investigación en esta línea de trabajo.

El resto del documento se organiza de la siguiente manera:

En el capítulo 2 se presenta el estado del arte necesario para comprender:

1. La situación actual en lo que respecta a leyes y normativas internacionales, regionales y en el Uruguay, que incluyan entre sus objetivos la accesibilidad educativa para personas con discapacidad.
2. Qué es Parálisis Cerebral, qué características tiene y cómo se clasifica, y qué tratamientos y avances se tienen al respecto.
3. Qué es el Diseño Universal, revisando las pautas, principios y cómo estos pueden aplicarse a la Educación.
4. Qué es la Interacción Persona Computadora, que modos de interacción existen, qué pautas de diseño universal puede aplicarse al diseño, en particular orientadas a niños y personas con discapacidad, y cómo se aplican a través de ejemplos de desarrollos existentes.
5. Qué es OLPC y Plan Ceibal, como atiende OLPC las necesidad de accesibilidad, que modelos de computadoras fueron entregadas por medio de estos planes en nuestro país, que especificaciones tienen estos modelos, como atiende OLPC y con qué actividades cuentan orientadas a la accesibilidad.

En el capítulo 3 se describe el contexto del proyecto que corresponde al contexto de la Escuela 200. Se detallan las patologías de los niños con especial énfasis en sus características de aprendizaje y comunicación. Se realiza también una descripción de los recursos didácticos que se utilizan .

El capítulo 4 presenta el proceso de análisis y diseño que se siguió, describiendo las actividades realizadas y los productos de las mismas, entre ellos: la selección de la herramienta a construir, la descripción de sus funcionalidades, el diseño de su GUI (Graphical User Interface o interfaz gráfica de usuario), la selección de herramientas a utilizar, y el diseño y arquitectura para la implementación del prototipo.

En el capítulo 5 se dan detalles sobre la implementación del prototipo, la estructura del código, ejemplos de particularidades de las herramientas seleccionadas y la descripción de algunos algoritmos.

En el capítulo 6 se describe el test de usabilidad realizado sobre y los resultados del mismo.

En el capítulo 7 se dan las conclusiones obtenidas y el trabajo a futuro.

Al final del informe se encuentran Anexos en donde se profundizan algunos de los temas tratados y se adjunta material utilizado.

Capítulo 2

Estado del Arte

2.1 Accesibilidad educativa de niños y niñas con discapacidad

Más de 1.000 millones de personas en el mundo entero viven con alguna forma de discapacidad. Casi 93 millones de esos discapacitados son niños los cuales padecen múltiples desigualdades y disponen de menos oportunidades para acceder a la educación de calidad que se imparte en contextos integradores [9]. La accesibilidad implica igualdad de oportunidades. La creación de entornos, programas y herramientas educativas accesibles hace posible que todas las personas, independientemente de sus capacidades, pueda acceder a la educación obligatoria y, posteriormente, a la formación escogida para su desarrollo e independencia personal.

Existen varios actores que trabajan sobre este tema intentando generar un marco donde la accesibilidad educativa pueda desarrollarse. De estos actores, los más importantes son las organizaciones internacionales y los estados, ya que cuentan con el poder de generar soluciones a largo plazo que impliquen cambios reales. Dentro de las organizaciones internacionales, dos de las más conocidas son UNESCO y UNICEF, ambas parte de las Naciones Unidas. Estas organizaciones promueven tratados y convenios internacionales que fomentan el desarrollo de políticas educativas inclusivas por parte de los gobiernos que transforman los sistemas educativos del país al que pertenecen, influyendo también en su región. Entre los otros actores encontramos diversas organizaciones no gubernamentales sin fines de lucro y fundaciones privadas de beneficencia como es por ejemplo la fundación Teletón.

En el **Anexo I** se profundiza sobre la situación a nivel de leyes y normativas a nivel internacional, de latinoamérica y nacional respecto a la accesibilidad educativa y la discapacidad.

2.2 Discapacidad Motriz: *Parálisis Cerebral*

La parálisis cerebral es la mayor causa de discapacidad motriz infantil [10]. Además la heterogeneidad de los problemas en las capacidades psicomotoras que engloba la parálisis cerebral, la hacen un objeto de estudio muy abarcativo dentro de lo que es el grupo de usuarios al que apunta el proyecto.

2.2.1 Definición y Diagnóstico

El término Parálisis Cerebral (PC) se utiliza para agrupar a un gran número de síndromes que se manifiestan como trastornos psicomotores permanentes y no progresivos. Son producto de lesiones

o malformaciones del Sistema Nervioso Central (SNC) que pueden originarse durante la gestación del feto hasta las primeras etapas del desarrollo que incluye los 3 a 5 primeros años de vida cuando el cerebro está inmaduro.

Los trastornos están por lo general acompañados de problemas sensitivos, sensoriales, cognitivos, de la comunicación, perceptivos, del comportamiento y trastornos convulsivos como la epilepsia. Es la discapacidad física más frecuente dentro en la niñez. Estos trastornos secundarios son en su mayoría efectos de los trastornos del movimiento y la postura, y constituyen los verdaderos limitantes de actividad [11].

La PC se diagnostica en los primeros años de vida y se identifica principalmente por los trastornos motores. Las manifestaciones clínicas pueden no estar presentes después de la lesión inicial. El diagnóstico se realiza después del período neonatal, aunque en aproximadamente una cuarta parte de los casos el daño neurológico se manifiesta ya en este período con la presencia de una agrupación de signos neurológicos conocida como encefalopatía neonatal. En general el daño de la función motora se manifiesta en los primeros 18 meses de edad posnatal. Es importante la identificación precoz de pacientes con PC para una orientación terapéutica temprana. El diagnóstico es clínico, no hay un estudio de laboratorio que diagnostique la PC.

2.2.2 Clasificación

Existen múltiples clasificaciones de PC que se realizan con distintos objetivos. La clasificación actual engloba múltiples ejes y resulta útil tanto para el análisis clínico como para estudios poblacionales [12]. Nos centraremos en las clasificaciones fisiopatológica, topográficas y según el grado de limitación funcional, realizando una asociación entre ambas y mencionando también los trastornos asociados.

2.2.2.1 Clasificación fisiopatológica

Alteraciones del tono

Hipertonía: aumento anormal en el tono muscular que produce un estado de rigidez en los músculos.

Hipotonía: disminución anormal en el tono muscular que produce que los músculos se mantengan en un estado de flacidez.

Trastornos del movimiento

Espasticidad: rigidez de movimientos, incapacidad para relajar los músculos.

Ataxia: mal equilibrio corporal, marcha insegura y dificultades en la coordinación y control de manos y ojos.

Discinesia: hace referencia a movimientos anormales e involuntarios. Entre ellos encontramos:

- **Distonía:** se caracteriza por contracciones musculares sostenidas que causan torceduras y movimientos involuntarios repetitivos.
- **Coreoatetosis:** se caracteriza por tener características de corea, que son movimientos involuntarios breves en la cara y en las zonas distales (las más lejanas al tronco como son las

manos y los pies) y características de atetosis, que son posturas retorcidas y de cambio frecuente en las zonas proximales (las más cercanas al tronco como por ejemplo el pecho).

2.2.2.2 Clasificación topográfica

Hemiplejía: afecta a uno de los dos hemicuerpos (derecho o izquierdo).

Diplejía: afecta los miembros superiores.

Paraplejía: afecta los miembros inferiores.

Cuadriplejía: los cuatro miembros afectados.

2.2.2.3 Clasificación según grado de limitación funcional

Para clasificar la limitación funcional vamos a utilizar la clasificación de la función motora gruesa (GMFCS) para la parálisis cerebral que define cinco niveles, los cuales difieren entre sí en aspectos significativos para la vida cotidiana. Las diferencias se basan en limitaciones funcionales, la necesidad de dispositivos auxiliares de la marcha (como por ejemplo muletas) o de movilidad sobre ruedas más que en la calidad de movimiento. Las características de cada nivel difieren también según la edad.

A groso modo lo que caracteriza cada nivel es la autonomía de desplazamiento. En el primer nivel camina sin restricciones, el segundo camina con limitaciones, el tercero camina utilizando un dispositivo manual auxiliar de la marcha, en el cuarto cuenta con auto-movilidad limitada, es posible que utilice movilidad motorizada y en el quinto nivel debe ser transportado en silla de ruedas.

Todas estas clasificaciones se conjugan para enmarcar distintas realidades que pueden llegar a ser muy heterogéneas y surgen de la necesidad de poder generalizar tratamientos, soluciones, entre otros, a partir de investigaciones y estudios sobre los diferentes casos.

2.2.3 Incidencia

Si bien existen múltiples combinaciones de las distintas clasificaciones definidas, hay algunas que se presentan con más frecuencias que otras, a continuación se presentan algunas de ellas[13]:

2.2.3.1 Parálisis Cerebral Espástica

Es la forma más frecuente, dentro de este grupo tenemos:

Tetraplejía Espástica

Es la más grave. Representa un 10 a un 40 por ciento de los casos de PC. Algunos pacientes presentan dimorfismo facial. Hay una espasticidad generalizada, en muchos casos tienden a opistótonos (postura anormal reconocible por la contracción continuada o rigidez de los músculos de tal forma que el cuerpo queda curvado hacia atrás en forma de C invertida con la cabeza y pies hacia atrás), pobre movilidad de las extremidades y con reflejos aumentados. Habitualmente son niños con un marcado retraso psicomotor, con microcefalia y alteraciones visuales y auditivas. Se pueden presentar variedades de tetraplejía como la hipotónica y la coreoatetósica. Según la GMFCs los niños con con tetraplejía espástica se encuentran en el quinto nivel.

Diplejía Espástica

Se presenta del 10 al 33 por ciento de los pacientes con PC. Las manifestaciones se producen mayoritariamente en las extremidades inferiores sobre las superiores, que incluso no son afectadas en la mayoría de los casos. Se detecta espasticidad con signos piramidales y problemas de coordinación de las extremidades inferiores con hipertonía de los músculos de la cadera y marcha en punta, además presenta posición de tijeras y pueden aparecer contracturas. En algunos casos hay crisis epilépticas, es frecuente el estrabismo y no suele afectar la capacidad intelectual.

Hemiplejía Espástica

Puede ser congénita o adquirida. Es más frecuente en varones. Representa entre el 23 y el 40 por ciento de los casos de PC, por lo que es la más frecuente. Casi siempre la parte más comprometida es la extremidad superior.

2.2.3.2 Parálisis Cerebral Discinética

Es la forma de PC que se relaciona más con factores perinatales, el 60 - 70 por ciento de los casos. Se caracteriza por un cambio brusco en el tono muscular, presencia de movimientos involuntarios. Se diferencian distintas formas clínicas: forma distónica, forma coreoatetósica y forma mixta que combina ambas y está asociada a la espasticidad.

2.2.3.3 Parálisis Cerebral Atáxica

Se caracteriza por un mal equilibrio corporal, dificultades en la coordinación y el control de manos y ojos.

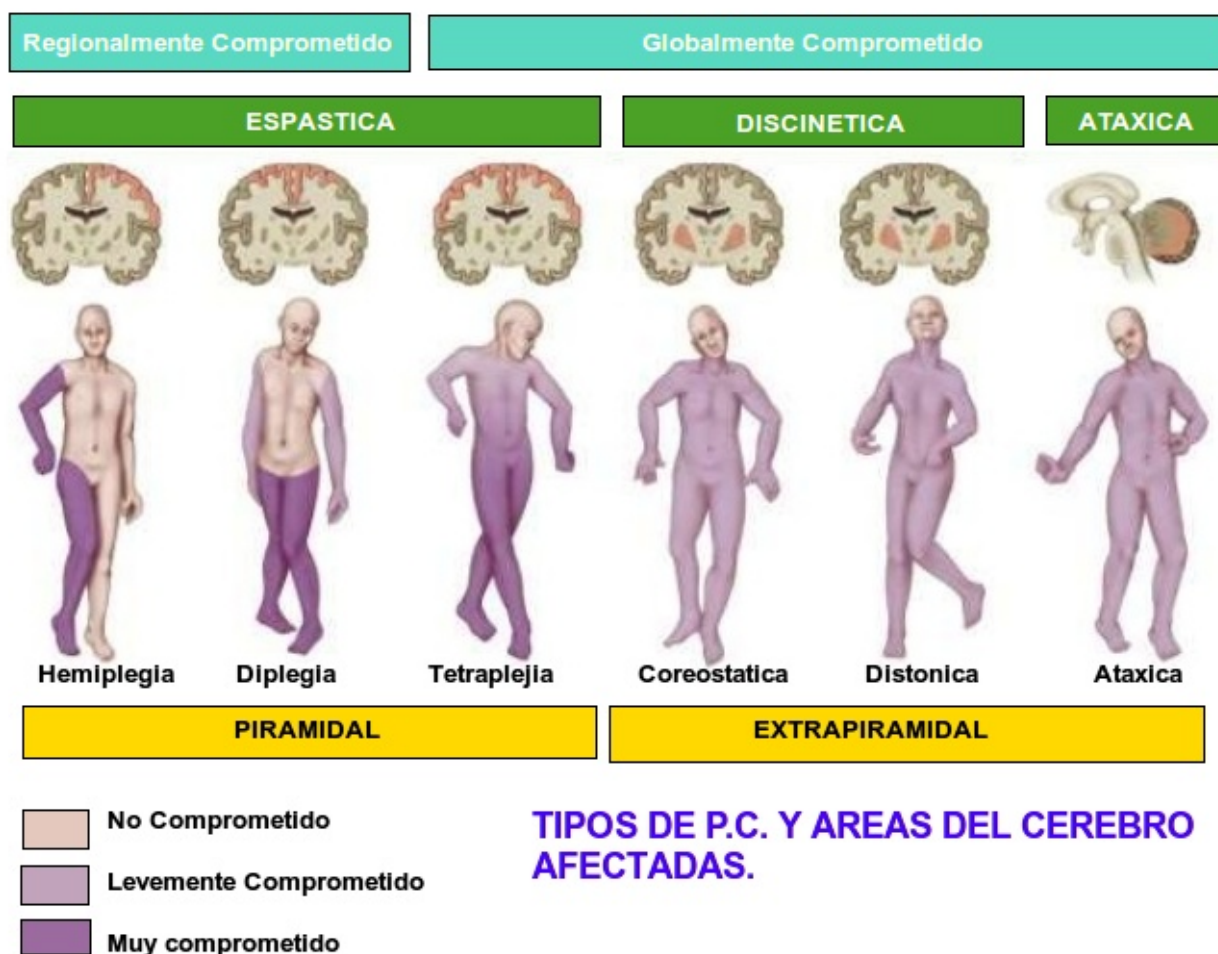


Figura 1: Tipos de Parálisis Cerebral

2.2.4 Dificultades asociadas

A continuación se describen las dificultades asociadas más habituales [11]:

Deficiencia Mental

El 70 por ciento de los casos la presenta. Existe una asociación entre el número de extremidades afectadas con el riesgo de presentar retraso mental. En niños con hemiplejía espástica el 60 por ciento tiene inteligencia normal mientras que en el caso de tetraplejía espástica del 70 al 80 por ciento tiene retraso mental. En algunos casos la PC puede ser tan sutil que se manifieste sólo con problemas de aprendizaje, visuales, de psicomotricidad y lenguaje leves.

Epilepsia

Aproximadamente la mitad de los niños con PC tienen epilepsia, muy frecuente en pacientes con tetraplejía (70%) y riesgo inferior al 20% en displéjicos.

Problemas de Lenguaje

Se pueden definir tres grupos:

1. Disfunción oral motora (producción de palabras)
2. Disfunción de procesamiento central (trastornos del lenguaje)
3. Disfunción auditiva (hipoacusia)

Se pueden presentar más de una de las alteraciones en una misma persona.

Parálisis Pseudobulbar

Es una de las manifestaciones más frecuentes en niños con PC. Se manifiesta por una succión débil o descoordinada, protrusión lingual (adelantamiento anómalo de la lengua al hablar) o problemas de deglución. Posteriormente presenta sialorrea (excesiva producción de saliva), desnutrición en algunos casos broncoaspiración con neumonía secundaria por el problema de la deglución. La salivación constante no es por exceso en la producción sino por no deglutir adecuadamente y ocasiona una humedad persistente, irritación en la piel de la cara y mala higiene bucal.

Trastornos Visuales

Son comunes los trastornos motores oculares como el estrabismo, nistagmus (movimiento constante, repetitivo e involuntario de los ojos) y problemas de refracción, ambliopía (disminución de la agudeza visual), defectos en campos visuales e incluso ceguera.

Problemas de Conducta

El déficit de la atención con hiperactividad e impulsividad son muy comunes en niños con PC y alto rendimiento intelectual. También niños con deficiencia mental presentan conductas destructivas, repetitivas y estereotipadas además de autoagresivas.

2.2.5 Tratamientos y Educación en niños y niñas con P.C.

Debido a las múltiples fases clínicas es necesario un equipo multidisciplinario que involucre: neuropediatra, fisioterapeuta, ortopeda, psicólogo, logopeda, pediatra de atención primaria y la colaboración de otros especialistas para lograr una valoración y una atención integral al niño con PC. El tratamiento debe de ser individual en función de la situación en que se encuentra el niño (edad, afectación motriz, capacidades cognitivas, patología asociada) y teniendo en cuenta el entorno familiar, social y escolar [13].

El tratamiento del trastorno motor se basa en cuatro áreas que pueden combinarse para obtener resultados: fisioterapia, ortesis (uso de férulas por ejemplo), fármacos y tratamiento quirúrgico.

2.2.6 Pronóstico

Las expectativas de vida de las personas con PC están relacionadas con la severidad y las complicaciones médicas asociadas a la misma, aumentando la mortalidad en los casos más comprometidos.

2.3. Diseño Universal y la Educación

2.3.1 Paradigma del Diseño Universal

Durante los pasados 35 años las definiciones sociales y los conceptos de discapacidad han cambiado radicalmente. La Organización Mundial de la Salud pasó de enfatizar un modelo médico a un modelo social, resultante de la interacción de la persona y el medio, y no solamente como algo propio de la persona. El reconocimiento del poder de los factores del medio para discapacitar o capacitar a una persona resalta el desafío para los diseñadores.

En 1985 durante la UNCORD (Convención sobre derechos de la personas con discapacidad de las Naciones Unidas) Ron Mace, un diseñador pionero del diseño universal y fundador del Centro de Diseño Universal, dijo en su discurso “*Universal Design means the design of products, environments, programs and services to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design*” (Diseño Universal significa el diseño de productos, ambientes, programas y servicios para que sean usados por todas las personas, para la mayor amplitud posible sin la necesidad de adaptación o diseños especializados). De aplicar este paradigma a todas las construcciones humanas, las discapacidades no serían tan discapacitantes y habría mayor inclusión e igualdad en ese aspecto.

El origen del paradigma actual del diseño universal puede rastrearse a dos fuentes distintas. Una se basa en medidas legislativas que incluyen requerimientos especiales para adecuarse a personas con discapacidad, principalmente afectando el entorno, y la otra en el mercado no regulado que responde a las necesidades de una sociedad envejecida.

En el año 1954 en los Estados Unidos, la suprema corte de justicia tomó la decisión, en el juicio de Brown contra el Consejo de Educación de Topeka, de que “separado no es igual” en un fallo histórico que declaró que las leyes estatales que establecen escuelas separadas para estudiantes de raza negra y blanca, negaban la igualdad de oportunidades educativas. Esta decisión de la suprema corte de justicia fue seguida por el movimiento de derechos civiles raciales de los sesenta, que logró llegar a la legislatura de derechos civiles incluyendo el derecho al voto y la no discriminación. Esto motivó en los setentas, un movimiento por los derechos de las personas con discapacidad que logró leyes federales que prohíben la discriminación por discapacidad. La legislación del año 73 incluyó requerimientos de accesibilidad a edificios construidos con fondos federales, luego esto pasó a incluir gran parte del entorno construido sin importar de donde vinieran los fondos. De todas formas el corolario de “separado no es igual” para la accesibilidad tenía tintes de una calidad similar a la estigmatización del asiento del fondo o del banco del burro, en donde si bien estaban en el mismo lugar se discriminaba igual, faltaba algo que aún sigue faltando que es un cambio social en ese sentido.

Para ver la otra fuente del paradigma actual tenemos que ir a Japón a la década de los 90, donde el diseño universal fue potenciado tanto por un interés económico como social, en respuesta al alto nivel de envejecimiento demográfico con el que contaba el país que originó un desafío tanto para la industria como para el gobierno japonés. Esto generó nuevas tendencias en el diseño de los inmuebles y los productos, así como las formas en las que se brindaban la atención hospitalaria y otros servicios tuvieron que ser repensados.

En el 2002 en la Conferencia Internacional sobre Diseño Universal se creó la Asociación Internacional de Diseño Universal, la IAUD (*International Association of Universal Design*), la

cual define en su página de trabajo como *“We seek for further progress and to make a comfortable living environment, and to lead Japan in disseminating information to the world. Through the products and services, we promote the establishment of the foundations of a society in which more people will feel comfortable to live.”* (Buscamos seguir avanzando para hacer un ambiente confortable donde vivir y de convertir a Japón en un difusor de información al respecto para el mundo. A través de los productos y servicios, promovemos el establecimiento de las bases de una sociedad en la que más gente se sienta cómoda para vivir). Esta corriente de pensamiento de un diseño universal que fomenta la IAUD puede ser vista ahora en todas partes del mundo. Los términos: diseño sin fronteras, diseño accesible, diseño inclusivo, diseño para todos y diseño universal ya son conocidos colectivamente como un concepto habilitador. El diseño universal refiere a un diseño que pueda ser usado por el rango más amplio de usuarios sin necesidad de adaptaciones [14].

En 1997 en el Centro de Diseño Universal dirigido por Mace, se desarrollaron 7 principios que articularon un proceso por el cual se define y evalúa la accesibilidad en el diseño de los elementos.

2.3.1.1 Los 7 principios del Diseño Universal [14]

1er Principio: Uso equivarable

El diseño es útil y vendible a personas con diversas capacidades.

Pautas para el Principio 1:

- Proporcione las mismas maneras de uso para todos los usuarios: idénticas cuando es posible, equivalentes cuando no lo es.
- Evite segregar o estigmatizar a cualquier usuario.
- Las características de privacidad, garantía y seguridad deben estar igualmente disponibles para todos los usuarios.
- Sea atractivo para todos los usuarios.

2do Principio: Uso flexible

El diseño se acomoda a un amplio rango de preferencias y habilidades individuales.

Pautas para el Principio 2:

- Ofrezca posibilidades de elección en los métodos de uso.
- Accesible y usable tanto con la mano derecha como con la izquierda.
- Facilite al usuario la exactitud y precisión.
- Adaptable al paso o ritmo del usuario.

3er Principio: Simple e intuitivo

Cómo se utiliza lo diseñado es fácil de entender, atendiendo a la experiencia, conocimientos, habilidades lingüísticas o grado de concentración actual del usuario.

Pautas para el Principio 3:

- Elimine la complejidad innecesaria.
- Sea consistente con las expectativas e intuición del usuario.
- Adaptable a un amplio rango de alfabetización y habilidades lingüísticas.

- Dispense la información de manera consistente con su importancia.
- Proporcione avisos eficaces y métodos de respuesta durante y tras la finalización de la tarea.

4o Principio: Información perceptible

El diseño comunica de manera eficaz la información necesaria para el usuario atendiendo a las condiciones ambientales o a las capacidades sensoriales del usuario.

Pautas para el Principio 4:

- Multimodal para presentar de manera redundante la información esencial (gráfica, verbal o de forma táctil).
- Proporcione contraste suficiente entre la información esencial y sus alrededores.
- Amplíe la legibilidad de la información esencial.
- Diferencie los elementos en formas que puedan ser descritas (por ejemplo, que haga fácil dar instrucciones o direcciones).
- Compatible con varias técnicas o dispositivos usados por personas con limitaciones sensoriales.

5o Principio: Con tolerancia al error

El diseño minimiza los riesgos y las consecuencias adversas de acciones involuntarias o accidentales.

Pautas para el Principio 5:

- Disponga los elementos para minimizar los riesgos y errores: elementos más usados, más accesibles; y los elementos peligrosos eliminados, aislados o tapados.
- Con advertencias sobre peligros y errores.
- Con características seguras de interrupción.
- Desaliente acciones inconscientes en tareas que requieren vigilancia.

6o Principio: Que exija poco esfuerzo físico

El diseño puede ser usado eficaz y confortablemente y con un mínimo de fatiga.

Pautas para el Principio 6:

- Permita que el usuario mantenga una posición corporal neutra.
- Utilice de manera razonable las fuerzas necesarias para operar.
- Minimice las acciones repetitivas.
- Minimice el esfuerzo físico continuado.

7o Principio: Tamaño y espacio para el acceso y uso

Que proporcione un tamaño y espacio apropiados para el acceso, alcance, manipulación y uso, atendiendo al tamaño del cuerpo, la postura o la movilidad del usuario.

Pautas para el Principio 7:

- Proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie.

- Alcance de cualquier componente sea confortable para cualquier usuario sentado o de pie.
- Que se acomode a variaciones de tamaño de la mano o del agarre.
- Que proporcione el espacio necesario para el uso de ayudas técnicas o de asistencia personal.

Estos 7 principios son conocidos a nivel mundial y fueron traducidos a 11 idiomas. El diseño universal es todavía joven en términos de la tecnología y generalmente no está incorporado en las políticas de desarrollo económico. Las premisas y aspiraciones del diseño universal siguen evolucionando y cada vez se ven más reflejadas en políticas, prácticas y ejemplos en todo el mundo. [14]

La morfología del diseño universal se nota con más claridad en tres áreas en particular en la Educación, Negocio y Social.

Si bien el área de negocio y social son relevantes en el diseño de herramientas inclusivas, nos vamos a centrar en el diseño universal en educación.

2.3.2 Diseño universal para la Educación

Al Dr. David Rose, neuropsicólogo y educador estadounidense, se le puede atribuir el título de padre o al menos de ser uno de ellos, de lo que es conocido hoy como Diseño Universal para el Aprendizaje o por sus siglas en inglés UDL (*Universal Design for Learning*). Rose cofundó en el año 1984 el Centro de Aplicación de Tecnología Especializada o CAST (Center for Applied Special Technology), que inició como un proyecto de tecnología asistiva para desarrollar y aplicar tecnología que habilitase a estudiantes con discapacidades acceder a una currícula impresa. A través de los años su interés pasó de la currícula a la transformación de los métodos y materiales educativos en general para que pudieran responder a la más diversa gama de estudiantes. Esto impulsó la creación del Centro Nacional de Diseño Universal para el Aprendizaje que es parte de CAST, donde se desarrollaron guías en tres áreas principales: representación, expresión y motivación del compromiso.

El Acta para las Oportunidades para la Educación Superior fue la primer legislación federal de los Estados Unidos en definir el UDL: “El Diseño Universal para el Aprendizaje es un marco de trabajo científicamente válido que - (A) provee flexibilidad en las maneras en que la información es presentada, en las maneras en que los estudiantes demuestran sus conocimientos y sus habilidades, y en las maneras en que los estudiantes se motivan; y (B) reduce los desafíos en la instrucción, provee espacios apropiados, apoyos y mantiene las expectativas de obtener grandes logros para todos los estudiantes, incluyendo estudiantes con discapacidades y estudiantes limitados en el uso del inglés”. [14]

Este marco de trabajo de trabajo se basa en los tres principios de CAST:

1. Múltiples formas de representación:

Incluyendo enseñanza multimodal, se basa en una mezcla de distintos medios de comunicación para transmitir los conceptos, como por ejemplo conferencias, podcasts, textos, recursos web como videos en youtube.

2. Múltiples formas de acción y expresión:

Dar a los estudiantes oportunidades para demostrar un aprendizaje a través de múltiples formas de evaluación tales como proyectos multimedia, documentos escritos y un proyecto

en lugar de un examen exhaustivo.

3. Múltiples formas de motivación:

Con el fin de maximizar el aprendizaje del estudiante, se enfatiza sobre la motivación dando la oportunidad a los estudiantes de interactuar y conectarse con el contenido de información para mejorar la retención.

Ver figura 2 para una información más detallada.



Figura 2: Diagrama de principios de UDL.

En lo que respecta a la tecnología, el diseño de ambientes inclusivos utilizando tecnología crea condiciones óptimas para adaptarse a las necesidades de los estudiantes que suelen ser heterogéneas. Cursos diseñados con principios del UDL y realizados utilizando tecnología asistiva, proveen la posibilidad a los estudiantes a acceder a los contenidos de múltiples formas y en diferentes momentos, dando acceso a que aquellos que no lo tenían potenciando sus oportunidades educativas. [14]

2.4 Interacción Persona Computadora

La Interacción Persona Computadora (IPC) es definida por el SIGCHI o *Special Interest Group in Computer Human Interaction* [15] que forma parte de la ACM o *Association for Computer Machinery* [16], como la disciplina relacionada con el diseño, evaluación e implementación de sistemas informáticos interactivos para el uso de seres humanos, y con el estudio de los fenómenos más importantes con los que está relacionado. En la IPC se interceptan varias disciplinas, algunas

de ellas son: Ingeniería de Software, Psicología, Sociología, Diseño, Inteligencia Artificial, Ergonomía y Programación.

La interacción entre personas y computadoras se realiza a través de una interfaz conocida como interfaz de usuario, donde ambas partes se transmiten información, órdenes y datos [17]. Las limitaciones tanto de expresión como de representación de esa interfaz definen su capacidad de comunicación, dado que lo que no se pueda expresar o representar queda fuera de la relación mutua. Si bien estos límites pueden deberse al estado actual del avance tecnológico, hay muchos casos que son producto de problemas de diseño e implementación [18].

Esta interacción ha evolucionado pasando a través de distintos paradigmas, de la compleja carga de programas por lotes en máquinas colosales en la década de los 50, al manejo sencillo de aplicaciones en teléfonos celulares inteligentes a través de pantallas táctiles en la actualidad.

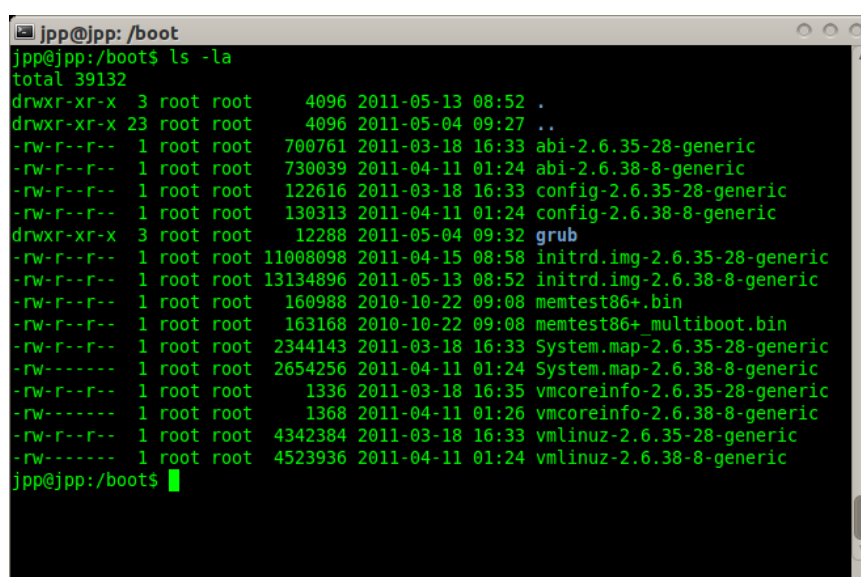
En las siguientes secciones se presentan los principales estilos y paradigmas de interacción que fueron surgiendo durante esta evolución.

2.4.1 Estilos de Interacción

Un estilo de interacción es un término genérico que se utiliza para agrupar las diferentes maneras en que los usuarios interactúan con la computadora, los estilos predominantes son [20]:

2.4.1.1 Línea de comandos

La línea de comandos fue el primer estilo de interacción y aún sigue siendo muy utilizado, sobre todo en sistemas Linux/Unix. Usualmente son utilizados por usuarios expertos quienes introducen comandos y parámetros que afectan la forma en que el comando es ejecutado. Ofrece acceso directo a funcionalidades del sistema y es muy flexible dado que las órdenes tienen normalmente una serie de parámetros que modifican su comportamiento. No obstante, esto implica una mayor dificultad de aprendizaje, una dificultad es que para poder escribir una orden es necesario tenerla memorizada, lo que hace difícil su uso por parte de usuarios novatos.



```
jpp@jpp: /boot
jpp@jpp:/boot$ ls -la
total 39132
drwxr-xr-x  3 root root    4096 2011-05-13 08:52 .
drwxr-xr-x 23 root root    4096 2011-05-04 09:27 ..
-rw-r--r--  1 root root  700761 2011-03-18 16:33 abi-2.6.35-28-generic
-rw-r--r--  1 root root  730039 2011-04-11 01:24 abi-2.6.38-8-generic
-rw-r--r--  1 root root  122616 2011-03-18 16:33 config-2.6.35-28-generic
-rw-r--r--  1 root root  130313 2011-04-11 01:24 config-2.6.38-8-generic
drwxr-xr-x  3 root root    12288 2011-05-04 09:32 grub
-rw-r--r--  1 root root 11008098 2011-04-15 08:58 initrd.img-2.6.35-28-generic
-rw-r--r--  1 root root 13134896 2011-05-13 08:52 initrd.img-2.6.38-8-generic
-rw-r--r--  1 root root   160988 2010-10-22 09:08 memtest86+.bin
-rw-r--r--  1 root root   163168 2010-10-22 09:08 memtest86+ multiboot.bin
-rw-r--r--  1 root root  2344143 2011-03-18 16:33 System.map-2.6.35-28-generic
-rw-r--r--  1 root root  2654256 2011-04-11 01:24 System.map-2.6.38-8-generic
-rw-r--r--  1 root root    1336 2011-03-18 16:35 vmcoreinfo-2.6.35-28-generic
-rw-r--r--  1 root root    1368 2011-04-11 01:26 vmcoreinfo-2.6.38-8-generic
-rw-r--r--  1 root root  4342384 2011-03-18 16:33 vmlinuz-2.6.35-28-generic
-rw-r--r--  1 root root  4523936 2011-04-11 01:24 vmlinuz-2.6.38-8-generic
jpp@jpp:/boot$
```

Figura 3: Ejecución de la orden `ls -la` en una línea de comandos linux, donde `ls` lista el contenido de los directorios y el parámetro `la` implica que muestre los detalles.

2.4.1.2 Formularios

Este estilo de interacción está dirigido a usuarios no expertos. Cuando aparecieron por primera vez toda la interfaz se basaba en formularios, a diferencia de la gran mayoría de software actual que mezcla distintos estilos de interacción. En aquel entonces, las pantallas eran diseñadas como formularios en los cuales la información podía ser ingresada en campos predefinidos.

La tecla TAB era (y aún sigue siendo) utilizada para cambiar entre los campos y la tecla ENTER para enviar el formulario. Por lo que originalmente no había necesidad de por un dispositivo de puntero como el ratón y la separación de datos en campos para la validación de la entrada.

Interfaces de formularios son útiles para tareas rutinarias o para tareas que requieren una gran cantidad de entrada de datos.

Figura 4: Ejemplo de *Fill Form* de aplicación para gestión de cuentas de tarjetas de crédito.

2.4.1.3 Menús y navegación

Un menú es un conjunto de opciones desplegadas en pantalla, donde la selección y ejecución de una (o más de una) de las opciones resulta en un cambio de estado de la interfaz. Utilizando un sistema basado en menús de selección, el usuario selecciona un comando de un conjunto predefinido de comandos arreglados en menús y observa el efecto. Si las etiquetas en los menús/comandos son bien comprensibles y están agrupadas de forma correcta, los usuarios pueden aprenderlas y recordar dónde encontrarlas rápidamente, completando sus tareas de forma sencilla al poco tiempo de uso. Para el ahorro de espacio de pantalla los menús están por lo general agrupados en menús pull-down o pop-up.

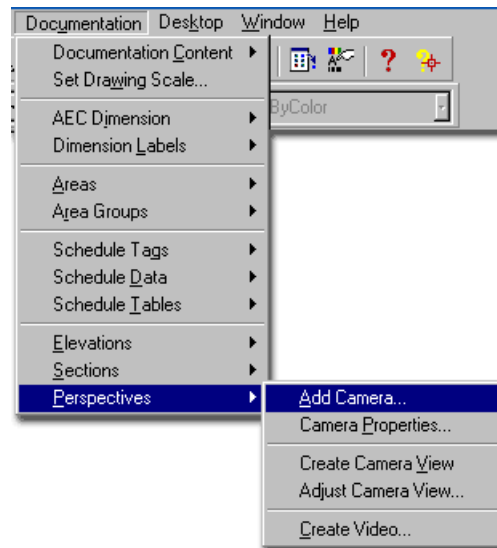


Figura 5: Ejemplo de pull-down

2.4.1.4 Manipulación directa

El estilo de manipulación directa captura la idea de la manipulación directa de objetos de interés, lo cual significa que los objetos de interés son representados como objetos distinguibles en la interfaz de usuario y son manipulados en un estilo directo, por ejemplo existe una representación gráfica de un archivo el cual el usuario puede agarrar y moverlo dentro de la representación gráfica de una carpeta.

Los sistemas con manipulación directa cuentan con las siguientes características:

- Visibilidad de los objetos de interés.
- Rapidez, reversibilidad de acciones incrementales.
- Reemplazo de una sintaxis de comandos compleja por una manipulación directa de los objetos de interés.

Estos sistemas presentan por lo general iconos representando objetos que pueden ser movidos por la pantalla y manipulados controlando un cursor con un ratón.

Entre las ventajas de un sistema de este estilo encontramos: leve curva de aprendizaje, mejoras en el trabajo multitarea y posibilidad de revertir acciones, algunas de sus desventajas son mayor consumo de recursos y la imposibilidad de representación directa de tareas en objetos concretos (por ejemplo para representar el concepto de buffer se utiliza el concepto de cortar y pegar).

La interfaz *Wimp*, es en la actualidad el entorno más común de manipulación directa. WIMP quiere decir Ventanas, Iconos, Menús y Apuntadores (*Windows, Icons, Menus and Pointers* en inglés).

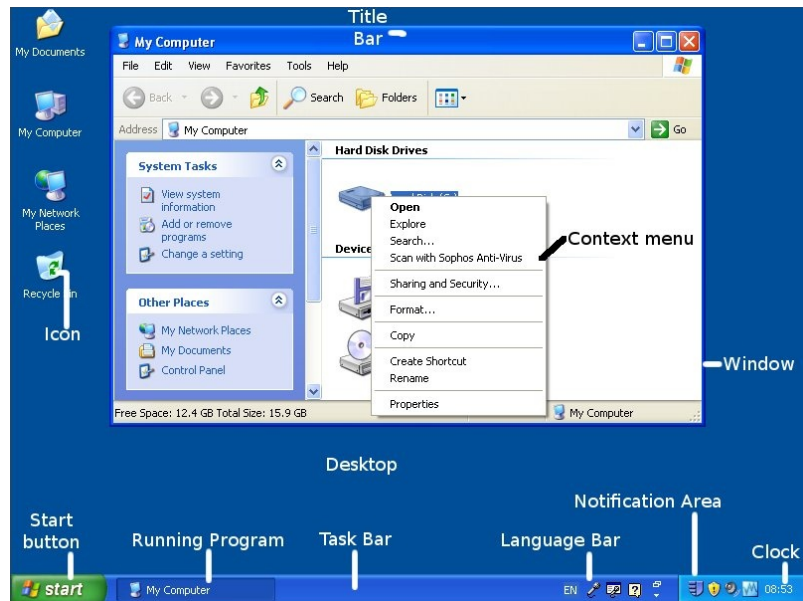


Figura 6: Descripción interfaz *Wimp* de *Windows XP*.

2.4.1.5 Interacción Asistida

Este estilo utiliza la metáfora del asistente personal o agente que colabora con el usuario en el mismo ambiente de trabajo y el usuario en lugar de dirigir la interacción como en la manipulación directa, trabaja en un entorno cooperativo en el que interactúa con agentes o asistentes realizando en conjunto las distintas tareas [19].



Figura 7: Ayudante de *Microsoft Office* incluido en *Office 97*

Agentes de Usuario

Los agentes de usuario son software cuya función es actuar de intermediarios inteligentes entre los usuarios y las computadoras. Fueron desarrollados como componentes autónomos entre las aplicaciones y los usuarios. Las interacciones con estos agentes suelen agruparse en lo que se denomina interacción asistida. Cuentan con una capacidad aprendizaje a partir de las acciones de

los usuarios de manera de luego poder dar consejos a los usuarios basándose en lo que aprendieron [19].

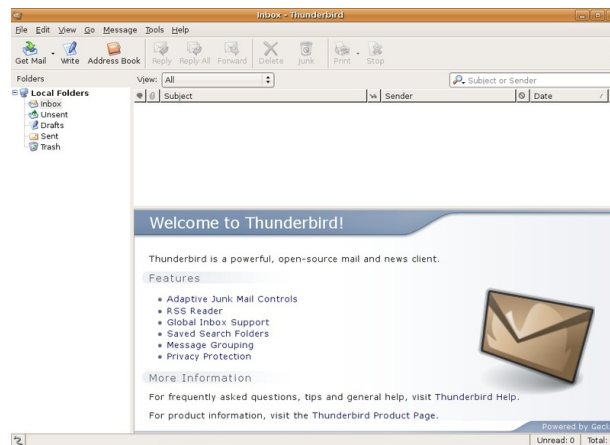


Figura 8: Thunderbird Agente de usuario para correo que sirve de intermediario entre el usuario y sus distintas casillas de correo.

Computación Emocional

Este estilo de interacción se basa en el reconocimiento de las emociones del usuario de manera de utilizarlas en la toma de decisiones y opciones que le puedan brindar al usuario.

Rosalind Picard del Massachusetts Institute of Technology (MIT) fundó esta línea de trabajo con su artículo de 1995 “Affective Computing”. Picard dirige actualmente el equipo de computación afectiva del MIT, en el sitio del equipo definen la motivación de su trabajo de la siguiente manera: “La emoción es fundamental para la experiencia humana, la cognición que influyen, la percepción y las tareas diarias tales como el aprendizaje, la comunicación y la toma de decisiones, incluso racional. Sin embargo, los técnicos han ignorado en gran medida la emoción y creado una experiencia a menudo frustrante para las personas, en parte porque afectar ha sido mal entendido y difícil de medir. Nuestra investigación desarrolla nuevas tecnologías y teorías que permitan avanzar en la comprensión básica del afecto y su papel en la experiencia humana. Nuestro objetivo es restaurar un equilibrio adecuado entre la emoción y la cognición en el diseño de tecnologías para hacer frente a las necesidades humanas.”

La computación afectiva debe realizar dos tareas de forma de lograr su cometido:

- El reconocimiento de emociones (y de expresiones emotivas) humanas por parte de una computadora, captando símbolos relacionados con la expresión de emociones y lograr interpretar estados emocionales en función de dichos signos.
- La simulación (o generación) de estados y expresiones emocionales con computadoras, tratando de simular procesos emocionales en base a ciertos modelos.

2.4.2 Paradigmas de Interacción

Los paradigmas de interacción son modelos de los cuales derivan los sistemas de interacción, son una abstracción de posibles modelos de interacción agrupados por características similares. A continuación se presentan los paradigmas actuales:

2.4.2.1 Computadora Personal

Es el paradigma dominante, el usuario interactúa con su computadora personal de manipulación directa. Esta interacción gracias al surgimiento de los dispositivos móviles puede darse casi en cualquier parte.

2.4.2.2 Realidad Virtual

El término realidad virtual (RV) refiere a tecnología cuyo propósito es el de generar espacios del mundo real o ficticios en los cuales los usuarios pueden sentirse dentro. Cuenta con la capacidad de producir en los usuarios la sensación física de encontrarse inmersos en una realidad completamente generada por computadoras. La forma primaria para generar inmersión es a través de experiencias visuales utilizando cascos con un visor especial que permiten una visión estereoscópica, pero también algunas simulaciones incluyen dispositivos para brindar mayor información sensorial como son parlantes o auriculares para generar el sonido ambiente, guantes para captar los movimientos de las manos y plataformas para captar desplazamiento en el ambiente. [19]



Figura 8: Realidad Virtual para prácticas de reanimación de la cruz roja, utiliza casco estereoscópico, auriculares y sensores para captar los movimientos de las manos

2.4.2.3 Computación Ubicua

El término de computación ubicua fue presentado por Mark Weisner en el año 1988, y refiere a la integración de la informática en los entornos de las personas, de forma que las computadoras y distintos dispositivos no se perciban como objetos diferenciados sino como parte propia del ambiente. [19] El término puede ser interpretado como “computadoras en todas partes”, Weisner en su artículo *The Computer of the 21st Century* publicado en 1991 dijo: “*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*”. (Las tecnologías más profundas son aquellas que desaparecen, se entretajan a sí mismas en el tejido de la vida cotidiana hasta que son indistinguibles de él) [20].

Tal vez una de las aplicaciones más conocidas de este concepto sea la Domótica en el que se integran sistemas a las viviendas capaces de automatizar los hogares a través de servicios de gestión energética, seguridad, comunicación, los cuales también se interconectan con redes para poder ser

controlados de forma remota. Esta integración se realiza de forma agradable tanto estéticamente como desde el punto de vista de su usabilidad.

Los dispositivos que se integran al entorno deben presentar ciertas características para permitir su correcta integración para la creación del entorno ubicuo, algunas de ellas son:

- Comunicación entre dispositivos: no solo con el usuario sino con los demás dispositivos.
- Memoria: deben tener capacidad de memoria y poder utilizarla para mejorar la interacción con los demás dispositivos.
- Sensibilidad al contexto: se deben adaptar a las situaciones, como por ejemplo preferencias de los usuarios, situaciones geográficas, hora del día, estación del año, etc.
- Reactivos: deben reaccionar cuando ocurren determinados eventos que pueden percibir en su entorno mediante sensores interactuando con los otros dispositivos, como por ejemplo prender el aire acondicionado si baja la temperatura a menos de una temperatura indicada.



Figura 9: Ejemplo de casa automatizada mostrando las áreas donde se encuentran los sistemas de automatización.

2.4.2.4 Realidad Aumentada

El término realidad aumentada (RA) refiere a tecnología cuyo propósito es el de añadir información a la realidad, o sea, aumentarla, utilizando elementos gráficos, sonoros o cualquiera que la tecnología disponible permita.

Ronald Azuma en el año 1997 definió que los sistemas de RA deben cumplir con las siguientes propiedades [21]:

- Combinación de objetos reales y virtuales en un entorno real.
- Ejecución en forma interactiva y en tiempo real.

- Registro y alineación de objetos reales y virtuales.

La combinación de objetos reales y virtuales se realiza a través de dispositivos que añaden información virtual a la información física, esta es la principal diferencia con la realidad virtual donde se sustituye todo el mundo físico por uno virtual.

Los dispositivos por lo general cuentan con una cámara para captar visualmente el entorno, la capacidad de reconocimiento de objetos y una pantalla para mostrar las información montada sobre la imagen ambiente, también pueden contar con auriculares para aportar sonido.

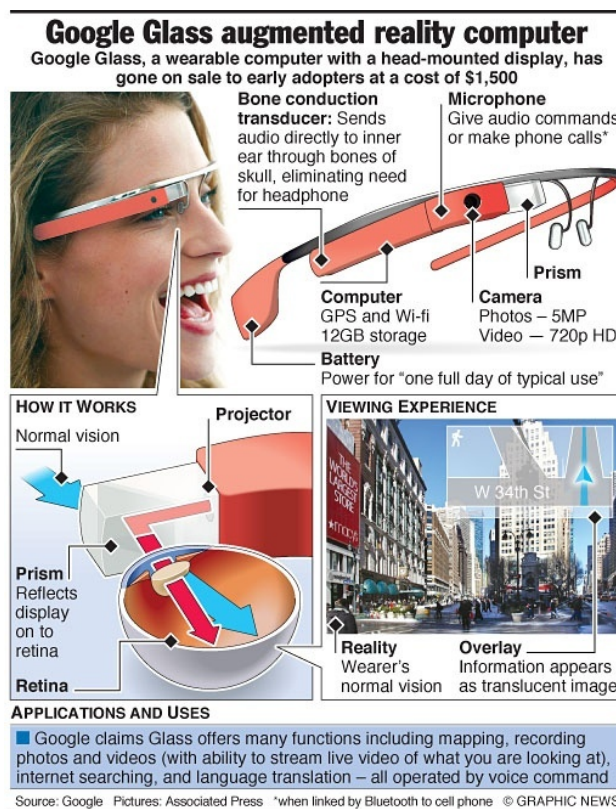


Figura 10: Descripción del dispositivo Google Glass, lentes de realidad aumentada de Google.

2.5 Diseño de Sistemas Interactivos

El desarrollo de sistemas interactivos y en particular su diseño, implica realizar un diseño centrado en el usuario, para ello es necesario incorporarlo en el proceso de desarrollo lo más posible, a través de una evaluación de usabilidad y accesibilidad durante cada una de las actividades que se realicen durante el proceso de desarrollo que se utilice.

Existen varios modelos de proceso definidos que introducen este concepto de evaluación continua del usuario, uno de ellos se describe a continuación:

2.5.1 Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad (MPIu+a)

Este modelo de proceso [21] consiste en una serie de actividades que se pueden dividir en tres categorías:

- Ingeniería de Software: incluye las acciones clásicas de un proceso de desarrollo Análisis / Diseño / Implementación / Instalación.
- Prototipado: actividades que permiten la posterior evaluación.
- Evaluación: engloba las actividades que se realicen para evaluar.

Es un modelo iterativo en el que se puede iterar hasta cumplir con determinadas metas marcadas.

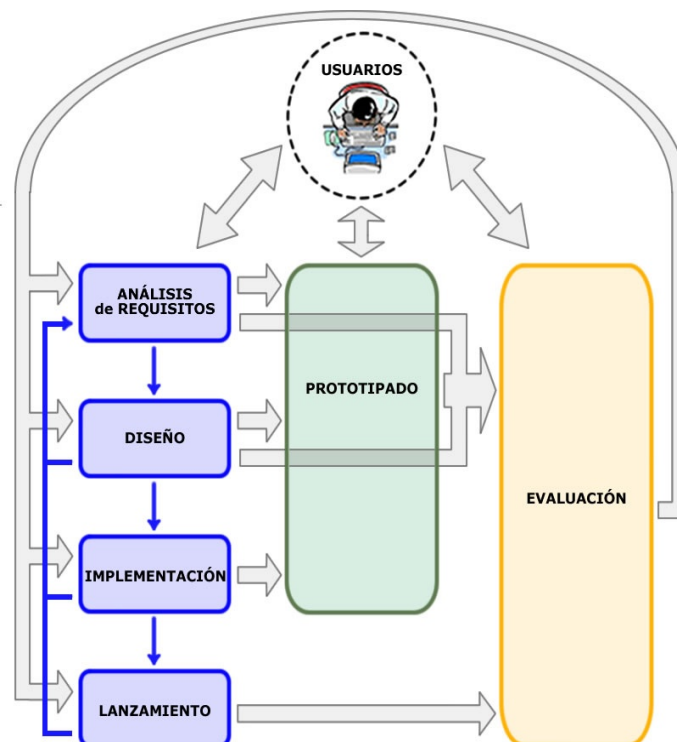


Figura 11: Esquema de MPIu+a donde se muestran las categorías de actividades, las flechas indican la secuencia y dan la idea de iteratividad del proceso.

A continuación se describen brevemente cada una de las actividades:

2.5.1.1 Análisis de Requisitos

Para la Ingeniería de Software es la actividad en donde se establecen los servicios que el sistema debe proporcionar y las restricciones bajo las que debe trabajar. Estos requisitos se dividen en:

- **Funcionales:** que describen las funcionalidades y servicios del sistemas
- **No funcionales:** que describen las restricciones para el sistemas (por ejemplo tiempo de respuesta) o para el proceso de desarrollo (tiene que funcionar en sistemas Android).

Esta actividad se divide en varias actividades:

- **Análisis Etnográfico:** hacer un análisis del grupo de usuarios, mediante la observación y descripción de lo que ellos hacen, cómo se comportan y cómo interactúan entre sí y con el medio.
- **Perfil de Usuario, actores y roles:** consiste en obtener una descripción de las características más relevantes de los usuarios del sistema y los actores intervienen, definir roles y asignarles subconjuntos de tareas para asociarlos con los usuarios.
- **Análisis contextual de tareas:** realizar un estudio de las tareas actuales de los usuarios, como las realizan, que patrones utilizan y llegar a especificar y entender sus objetivos.
- **Plataforma:** relevar qué plataforma se utilizará el sistema, esto define posibilidades y restricciones tecnológicas.
- **Objetivos de Usabilidad:** que objetivos de usabilidad debe tener el sistema, entre ellos pueden encontrarse: facilidad de aprendizaje, consistencia, flexibilidad, robustez, recuperabilidad, tiempo de respuesta, adecuación a las tareas y disminución de la carga cognitiva.
- **Descripción de Requerimientos:** describir los requerimientos tanto funcionales como no funcionales, de manera de que queden registrados correctamente y puedan ser evaluados.

2.5.1.2 Diseño

El diseño es donde se definen uno de los aspectos fundamentales del sistema interactivo dado que se centra en el diseño de la interfaz de usuario y cómo serán las interacciones, para luego definir lo que es el diseño de la arquitectura y lógica de la aplicación.

El diseño consta de las siguientes actividades:

- **Análisis de Tareas:** basándose en los requisitos obtenidos durante el análisis, se analizan las tareas del usuario de manera organizarlas de forma racional. Algunos de los métodos existentes para realizar el Análisis de Tareas: Análisis Jerárquico de Tareas o HTA (*Hierarchical Task Analysis*), GOMS (*Goals Operations Methods Selection*), KLM (*KeyStroke Level Mode*), TAG (*Task Action Grammar*), UAN (*User Action Notation*), CTT (*ConcurTaskTrees*) y GTA (*Groupware task Analysis*). [18]
- **Modelo Conceptual:** partiendo también del análisis, se genera a un nivel primario las primeras alternativas de diseño, que incluye las principales pantallas y sus caminos de navegación.
- **Definición de Estilo:** el propósito de esta actividad es el de definir un estilo que garantice la

coherencia visual y funcional de toda la aplicación. Es recomendable documentar bien el estilo para que sirva de guía para todo el sistema. Se debe tener cuidado en definir estilos en los casos en que se deben cumplir con estándares y en la utilización colores.

- **Diseño Detallado:** esta actividad se realiza luego de las actividades anteriores y una vez que se hayan realizado y probado prototipos a partir de ellas (al menos en una oportunidad).

2.5.1.3 Prototipado

El prototipado engloba todas las herramientas y actividades que permiten realizar simulaciones de lo que se está desarrollando durante su construcción. El modelo no marca una pauta sobre en qué situaciones se debe recurrir al uso de determinada o determinadas técnicas de prototipado, dando un alto grado de libertad de decidir cuando y como aplicarlas.

Algunas técnicas de prototipado son:

- **Prototipos de Papel:** utiliza materiales básicos que permitan describir un diseño en papel. Permite con gran velocidad y de forma económica, realizar una descripción a través del dibujo (sin entrar en grandes detalles estéticos) de interfaces para evaluarlas.
- **Storyboards:** consisten de una serie de viñetas que muestran la evolución de las distintas situaciones por las que puede pasar un usuario mientras interactúa con el sistema.
- **Herramientas de Diagramación:** dentro de estas herramientas se encuentran: la narrativa que consiste en contar en modo de historia la interacción del usuario, *flowchart* que consiste en una representación gráfica de las acciones y decisiones de la narrativa, y por último el texto procedural que es una descripción paso a paso del diálogo entre el usuario y el sistema.
- **Prototipos de Software:** son primeras versiones de ciertas funcionalidades desarrolladas con la tecnología escogida para desarrollar la aplicación.

2.5.1.4 Evaluación

Los prototipos no tiene sentido si no se someten a una evaluación. El prototipo es una herramienta de gran utilidad que permite la participación del usuario en el desarrollo y la evaluación del sistema desde etapas tempranas de su desarrollo.

- **Inspección:** agrupa un conjunto de métodos en los que expertos conocidos como evaluadores que explican el grado de usabilidad de un sistema basándose en la inspección de su interfaz. Algunos de estos métodos son:
 - **Heurística:** consiste en validar la conformidad de la interfaz con uno principios reconocidos de usabilidad (la “heurística”) mediante la inspección de varios evaluadores expertos [23]. Existe un conjunto de reglas heurísticas conocidas como las “10 reglas heurísticas de la usabilidad” el cual fue confeccionado a partir del análisis de 249 problemas de usabilidad [24].
 - **Recorrido de la usabilidad plural:** consiste en la evaluación de prototipos en papel por tres participantes y en un debate al final entre ellos.
 - **Recorrido Cognitivo:** se realiza por medio de la exploración del prototipo y se centra en evaluar la facilidad de aprendizaje del sistema.
- **Indagación:** agrupa el conjunto de métodos en lo que se obtiene el conocimiento sobre la usabilidad de la interfaz reflexionando sobre ella o por conjeturas y pruebas. Algunos de estos métodos son:
 - **Observación de Campo:** consiste en observar a los usuarios para entender cómo realizan sus tareas y que clase de modelo mental tienen sobre ellas. La información

- recabada luego es completada con preguntas y/o entrevistas.
- **Focus Group:** es una técnica de recolección de datos donde se reúne de 6 a 9 usuarios para discutir aspectos relacionados con el sistema. Se debe contar con un moderador, una lista de aspectos a discutir y una manera de registrar la discusión [18].
- **Entrevistas:** consta de entrevistar a los usuarios respecto de su experiencia con el sistema.
- **Logging:** se basa en grabar o recoger todas las actividades que realiza el usuario con el sistema para su posterior análisis.
- **Cuestionarios:** es menos flexible que la entrevista pero permite llegar a un grupo más numeroso.
- **Test:** agrupa a los métodos de usabilidad en los que los usuarios representativos realizan tareas utilizando el sistema mientras son observados y registrados, para que luego evaluadores puedan analizar la usabilidad del sistema a partir de la información recabada. Algunos de estos métodos son:
 - **Medida de las Prestaciones:** consiste en realizar el test con usuarios reales realizando labores habituales también reales.
 - **Thinking Aloud:** en este test se les pide a los usuarios que expresen en voz alta sus pensamientos, sentimientos y opiniones mientras interactúan con el sistema [18].

2.5.1.5 Implementación

Es donde se realizan todas las tareas de programación del sistema. Implica haber escogido los lenguajes de programación que mejor se adapten al proyecto. Esta actividad no requiere una descripción detallada dado que es exactamente igual la que se realiza siguiendo un proceso de ingeniería de software clásico. El *MPIu+a* no afecta esta actividad.

2.5.1.6 Lanzamiento

Es el momento en que el sistema se libera a los usuarios finales del sistema, y donde se recibe la reacción del usuario y se puede evaluar el éxito del producto.

Por lo general cuando un sistema es puesto en producción, se define un período en de pruebas, en el que se recoge esa reacción del usuario a la que se le suele llamar *User Feedback* [18] a través de cuestionarios o entrevistas para registrar impresiones, o utilización de herramientas para el registro sugerencias de mejora y reportes de error como *Mantis Bug Tracker* [25] o *Bugzilla* [26].

A partir del User Feedback se realizan las modificaciones que se consideren necesarias al sistema, dejándolo nuevamente en el período de pruebas.

2.5.2 Diseño orientado a niños

Los niños tienden a ser creativos, curiosos y muy inteligentes. Estas cualidades deben ser tomadas en cuenta al momento de diseñar interfaces para niños de manera de explotarlas y potenciarlas a través de un diseño que permita la utilización correcta y sencilla de las tecnologías para las cuales se está diseñando.

El proceso de diseño de interacción para niños no debería ser muy diferente del proceso de diseño

de interacción para adultos, pero los métodos para realizar el relevamiento de requerimientos, la implementación y la evaluación necesitan adaptarse para poder atender a necesidades particulares de este grupo.

A continuación se presentan algunas de las adaptaciones que deben ser consideradas al realizar las actividades durante el proceso.

2.5.2.1 Relevamiento de Requerimientos

Los sistemas desarrollados para niños son por lo general divertidos y educativos, pero saber cuáles son las necesidades específicas de los niños no es tarea sencilla. Las formas tradicionales de relevar los requerimientos de los usuarios (por ejemplo, las entrevistas y encuestas) pueden no ser apropiadas para los niños dadas sus limitadas capacidades de lenguaje y de atención. A su vez, los niños pueden no saber qué necesitan [14].

En estos casos es útil:

- Consultar a usuarios indirectos como pueden ser padres o maestros. Interactuando con los padres y los maestros, el diseñador puede tomar una noción de las necesidades de los niños, como por ejemplo darse cuenta que la necesidad de niño no es tener un juego de matemáticas para resolver multiplicaciones per se sino como una forma didáctica de memorizar la tablas de multiplicación.
- Observar a los niños en sus ambiente natural como durante el juego o en una clase puede proveer experiencia en los tipos de actividades que le gustan a los niños y también en los modos en que interactúan con diferentes objetos para buscar soluciones de diseño[14].

2.5.2.2 Diseño de Interacción

A continuación se presentan algunas pautas importantes a tener en cuenta al diseñar interacciones orientadas a niños [61] :

Diseñar contenidos apropiados para las distintas edades: Es importante diseñar actividades acordes al desarrollo cognitivo del grupo de usuarios objetivo.

Entender los modelos mentales de los niños: Es importante recordar que los niños no tienen las mismas experiencias de vida que los adultos por lo tanto no tienen los mismos modelos mentales con lo que respecta a comprender el mundo que los rodea. Por ejemplo, un niño puede estar familiarizado con animales que forman parte de su hábitat como pueden ser el perro y el gato, pero pueden no conocer como se ve un pingüino ni qué sonido hace.

Usar un lenguaje apropiado: Debido al desarrollo de las habilidades del lenguaje en los niños, es importante que el lenguaje usado sea entendible para el grupo objetivo. Esto incluye dar instrucciones en un lenguaje claro y simple, utilizando palabras que estén en el vocabulario corriente de los niños.

Identificar restricciones ergonómicas: como las actividades son por lo general desarrolladas para plataformas diseñadas para el uso de adultos, es importante tener en cuenta las complicaciones a las que los niños se pueden enfrentar cuando interactúan con diferentes dispositivos, como por ejemplo dispositivos móviles grandes que no puedan manipular fácilmente.

No utilizar tamaños de fuente muy pequeños: Esta pauta aplica a todos, incluyendo niños.

Usar un buen contraste de color: Esta pauta también aplica a todos, incluyendo niños. En

particular, como los niños responden mejor a interfaces coloridas mantener un adecuado contraste exigirá un balance entre estos dos requerimientos.

Utilizar dispositivos de entrada con keypads grandes: Como los niños no tienen necesariamente habilidades motoras finas requeridas para hacer movimientos muy precisos ni exactos, es conveniente utilizar dispositivos de entrada que cuenten con keypads grandes como por ejemplo las pantallas táctiles.

Apuntar y hacer click vs. Drag and Drop: Investigaciones han encontrado que los niños tienden a realizar actividades del modo apuntar y hacer click, más rápidamente y de forma más precisa que actividades drag and drop. Con esto en mente, los diseñadores deben tratar de aplicar estilos de interacción adecuados de acuerdo a la edad del grupo objetivo.

2.5.2.3 Evaluación

Como cualquier otro proceso de diseño centrado en el usuario, la evaluación es crítica para entender cómo los usuarios interactúan con un producto con el fin de que pueda ser mejorado. Sin embargo, los métodos de evaluación de interacciones con adultos no necesariamente funcionan con niños. Por ejemplo, los niños suelen dar respuestas más superficiales si no han entendido la pregunta. También son muy susceptibles a las sugerencias del evaluador y tiene una reducida capacidad de recordar eventos.

Algunas formas de superar estas complicaciones son:

- Realizar las evaluaciones en los ambientes naturales de los niños. La observación participativa es particularmente útil en este contexto ya que permiten al evaluador observar el comportamiento de los niños sin la necesidad de confiar en el recuerdo de los eventos.
- Evitar preguntas sugerentes y terminología específica para prevenir respuestas superficiales.
- Usar preguntas abiertas en lugar de preguntas específicas para incrementar la confiabilidad.
- Entrevistar otras personas (por ejemplo, padres y maestros) que puedan aportar diferentes visiones de las interacciones de los niños con el diseño y otra información útil.
- Hacerla divertida de manera de atraer la atención de los niños e incentivar su participación.

2.5.3 Diseño orientado a niños y niños con parálisis cerebral.

Un aspecto muy interesante desde el punto de vista del diseño de interfaces y del desarrollo de aplicaciones, es la posibilidad de crear software que permita adecuarse a las demandas de la heterogeneidad de usuarios en cada momento. Una visión evolutiva del software implica que los requisitos que habían sido previamente identificados para un tipo de usuario pueden variar. La adaptabilidad a estos cambios requiere un diseño que contemple tanto los requisitos iniciales como los cambios que se produzcan a medida que las necesidades de los usuarios vayan evolucionando con el uso del sistema, realizando así diseños orientados a perfiles. Los usuarios se agrupan en perfiles, en los cuales pueden ser miembros únicos, a cada perfil se le asocia un modelo conceptual (objetos, acciones, modos de interacción) atendiendo a sus habilidades físicas y cognitivas. En este sentido, es relevante modelar el propio entorno del usuario, ya que este constituye una capa de contexto en el cual tiene lugar la comunicación y la actividad de la persona. El término de software no anticipativo se usa cuando la adaptabilidad no está previamente programada, los cambios en el perfil de usuario o en el contexto no son conocidos a priori y suceden en tiempo real [14].

Los sistemas aumentativos y alternativos pueden concebirse bajo esta perspectiva. Los sistemas con ayuda utilizan un artefacto auxiliar para la comunicación como pueden ser los tableros comunicadores basados en lenguajes simbólicos. Sin embargo, un problema que permanece constante, es la necesidad de adaptar tanto el tipo de comunicador como el modelo de lenguaje usado (conceptos, representación, complejidad) a las necesidades de cada usuario. Esto significa que cada comunicador se debe programar para cada caso y situación, lo que implica un esfuerzo considerable en caso de que se quiera realizar un uso extendido y generalizado.

2.5.4 Desarrollos realizados con un diseño orientado a niños con Parálisis Cerebral

A continuación se presentan algunos desarrollos cuyos diseños permiten su utilización por parte niños con PC o con discapacidades cuyas patologías sean similares las de la PC.

SwitchXS

SwitchXS [28] es una aplicación para Mac que provee acceso virtual completo a sistema operativo de Mac, el Mac OS X, y a todas las aplicaciones estándar, a personas que puede utilizar uno o más switches, como por ejemplo un switch sip-and-puff que permite generar eventos mediante aire a través del soplido y la succión.



Figura 12: SwitchXS logo



Figura 13: Switch sip-and-puff utilizado para manejar una silla de ruedas eléctrica.

Descripción

SwitchXS ofrece una completa emulación de ratón y teclado a través de un modo de escaneo con barrido. Con evento generado por el switch el usuario selecciona una acción, como por ejemplo “mover el cursor arriba” o “típear B”, desde un menú que se escanea. *SwitchXS* es usado comúnmente por personas con problemas motores que le limitan el movimiento.

Provee un número predefinido de paneles escaneables de los cuales los usuarios pueden elegir mover el cursor, hacer *click* y tipear en todas las aplicaciones estándar de Mac. También permite al usuario controles avanzados a través de funcionalidades como escucha de texto, repetición de teclas, velocidad de escaneo y ciclos, velocidad y movimiento del ratón, y *SmartTransparency* que hace que los paneles de escaneo parcialmente o completamente invisibles cuando no se está realizando escaneo alguno permitiendo un óptimo uso de la pantalla.

SwitchXS trabaja con cualquier dispositivo de entrada para las computadoras Mac, como pueden ser gamepads, joysticks, teclados, ratones o trackball. También trabaja con cualquier interfaz que simule alguno de los dispositivos mencionados.

SwitchXS incluye *LayoutKitchen2*, un editor que permite a los usuarios crear sus propios paneles escaneables, incluyendo paneles cuyas teclas son totalmente personalizadas con una imagen y una acción creada por el usuario. Para la predicción de palabras, *SwitchXS* puede ser combinado con *KeyStrokes* otro teclado que cuenta con un excelente predictor de palabras.

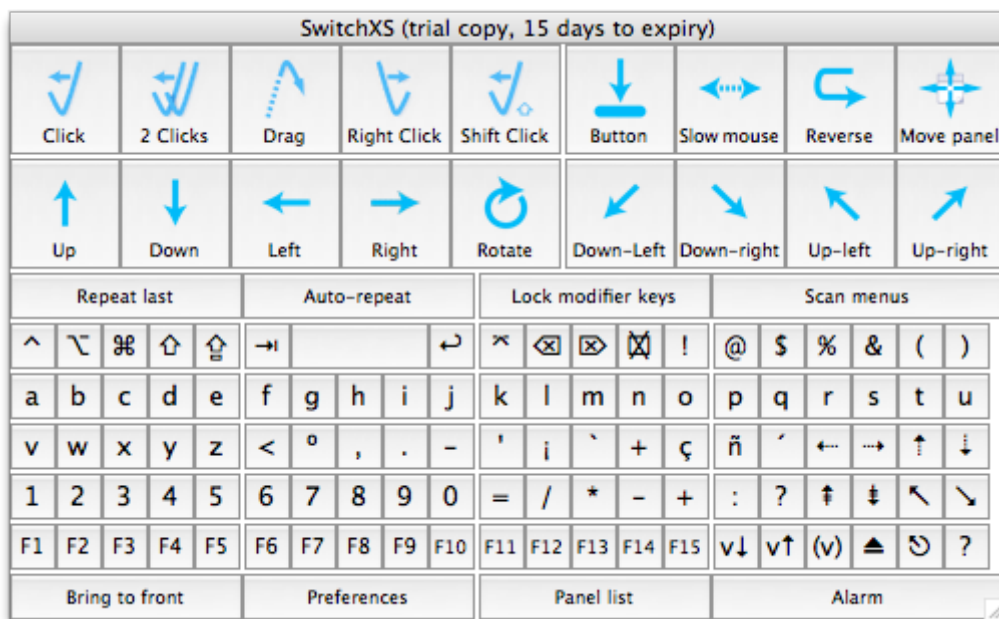


Figura 14: SwitchXS menú escaneable con barrido.

Funcionalidades Clave

- Completa emulación de ratón y teclado.
- Trabaja igual en todas los Mac OS X y en todas las aplicaciones clásicas incluso con configuraciones de múltiples monitores.
- Soporta múltiples switches con métodos de escaneo configurables, incluyendo escaneo inverso y escaneo por pasos.
- Provee métodos de escaneo de un solo botón con barrido y selección automáticos.
- Incluye tecnología SmartTransparency, la cual hace de la ventana de escaneo volverse transparente cuando no está siendo usada y la hace reaparecer de forma automática cuando se presiona el switch.
- Soporta combinaciones de teclas, teclas muertas y combinaciones tecla/click para realizar modificaciones.
- Provee señales de audio y habla.
- Ofrece múltiples paneles de escaneo predefinidos que cambian de tamaño de forma interactiva.
- Ofrece la posibilidad de configurar la velocidad de escaneo y los ciclos , velocidades diferenciadas de escaneo del menú, del ratón y reduce de forma automática la velocidad del ratón para adaptarse las preferencias de cada usuario.
- Trabaja con un amplio rango de dispositivos de entrada.
- Permite disparar AppleScripts para realizar múltiples acciones con un solo botón.
- Incluye LayoutKitchen2, que permite a los usuarios diseñar sus propios paneles escaneables con muchas funcionalidades avanzadas, incluyendo botones coloridos y texto así como también imágenes.
- Tiene acceso a la predicción de palabras al ser combinado con KeyStrokes, lo cual permite una mejor muy significativa en la velocidad de escritura lo cual reduce el esfuerzo.

Especificación Técnica

SwitchXS corre sobre Mac OS X versión 10.4 y superiores. Requiere una resolución de pantalla mínima de 800 x 600.

Para utilizar Mac OS X versión 10.4, se necesita tener una computadora Mac con:

- Un procesador PowerPC G3, G4 o G5.
- Al menos 256 MB de RAM.
- Una unidad de DVD para la instalación.
- Al menos 3 GB de espacio en el disco duro.

TobiiC12/C15 con el módulo Tobii Ceye

Descripción

La Tobii C12 [29] desarrollada por Tobii Technology [30] es una tablet adaptable a un amplio rango de habilidades comunicativas y métodos de acceso. Contiene software de comunicación desarrollado por Tobii que permite comunicación basada tanto en texto como en símbolos, así como también con email, internet, telefonía celular y capacidades ambientales como controlar la TV, DVD y otros dispositivos caseros.



Figura 15: Logo de Tobii, una de las empresas líderes en desarrollo de tecnología asistiva con la utilización del seguimiento de la mirada.



Figura 16: Tobii C12

El módulo Tobii Ceye [31] es una unidad de control ocular la cual está diseñada para encajar debajo de la Tobii C12. Al mirar la pantalla, el usuario puede controlar la pantalla con el movimiento ocular y luego hacer click al parpadear, mirando la pantalla al mismo lugar por un

tiempo determinado o usando un switch. El control ocular es utilizable por la mayoría de niveles cognitivos, dependiendo del software que está siendo usado. Los usuarios de este tipo de dispositivo no necesitan utilizar ningún otro equipamiento para acceder a la computadora, se colocan frente a la computadora y ésta traduce los movimientos oculares en movimientos del ratón en la pantalla.

La calibración con el Tobii CEye debería necesitar solo de una instancia para la gran mayoría de usuarios. Los tres botones de acceso rápido del lado de la Tobii C12 pueden ser habilitados para permitir el acceso rápido al estado del seguidor de la mirada, así como también a funciones de calibración y pausa. Dentro del software de comunicación de Tobii hay un función de pausa que permite a los usuario pausar el control por medio de la mirada.

El Tobii Ceye puede operar razonablemente bien incluso cuando el usuario tiene algún temblor ocular o movimientos de cabeza debido a que cubre una área de seguimiento bastante grande. Por lo que no es necesario que la cabeza del usuario esté totalmente quieta, esto es muy útil para usuarios con parálisis cerebral que sufren de movimientos involuntarios de cabeza. Si la cabeza se mueve fuera del área de seguimiento, el Tobii Ceye captará el movimiento nuevamente una vez que el usuario reingrese en área sin la necesidad de recalibración. El Tobii Ceye trabaja muy bien bajo varias condiciones de luz y en usuarios que usan lentes, ya sea de montura o de contacto.

Funcionalidades Clave

La Tobii C12 incluye el paquete de software Tobii Communicator 4 Standard Edition el cual permite tanto comunicación basada en texto como en símbolos. Dentro del paquete se encuentra:

- Sono Lexis, un vocabulario basado en símbolos que provee una progresión en tres etapas.
- Sono Key que provee un acceso basado en texto.
- Sono Scribe que provee una comunicación basada en la escritura.



Figura 17: Pantalla de Tobii Communicator 4 para expresar sentimientos.

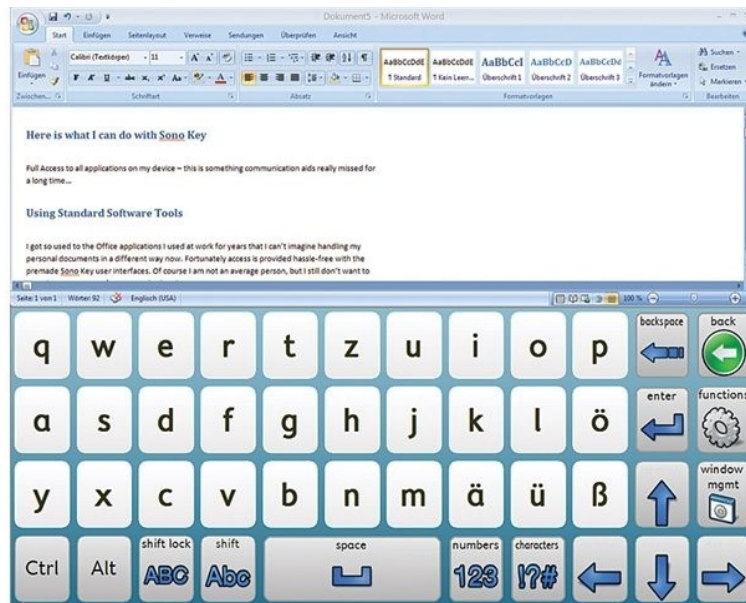


Figura 18: Teclado Tobii Sono Key



Figura 19: Pantalla de Tobii Sono Scribe

- Permite guardar múltiples perfiles de usuario, lo cual es útil para salones de clase.
- Capacidad de dictado: el paquete básico de la Tobii C12 incluye voces artificiales en un lenguaje.
- Opciones de acceso: la Tobii C12 está diseñada para la operación directa mediante la pantalla táctil de 12 pulgadas la cual es muy resistente. También es accesible a través de uno o dos switches externos como pueden ser un joystick o ratón.
- Cuenta con controles infrarrojos los cuales permiten controlar dispositivos del hogar como una televisión o un reproductor de dvd.
- Las dos baterías que trae pueden ser intercambiadas sin tener que apagar el aparato, y esto

- da alrededor de 4 horas de uso continuo.
- La Tobii C12 puede ser montada y usada sobre una silla de ruedas. Existen escritorios integrados para distintos modelos de equipamientos para sillas de ruedas.
- Tobii C12 cuenta con una cámara integrada que permite a los usuarios capturar imágenes para utilizarlas en sus pantallas de comunicación.

Especificación Técnica

- Dos micrófonos integrados.
- Procesador Intel Core Duo U2500 con memoria de 2GB DDR2.
- Microsoft Vista Home Premium.
- 60Gb de disco duro.
- Tarjeta sd de 1GB.
- Conectores: 2 x USB 2.0, 1 x Micro USB, 1 x RJ45, 2 x switch, 1 x auriculares (estéreo), 1 x DC in.
- WLAN opcional.
- Bluetooth opcional.
- GSM opcional.
- Cámara de 640x480 pixels.

iFreeTablet

Descripción

iFreeTablet [32] es una tablet pc de bajo costo, la cual cuenta con un diseño accesible que le permite ser utilizada por personas en situación de dependencia (niños, personas en riesgo de exclusión, discapacitadas o mayores).



Figura 20: Logo de iFreeTablet

Es un proyecto sin fines de lucro del grupo de investigación EATCO [33] de la Universidad de Córdoba de España, la fundación Fundación Red Especial España (FREE) y la Asociación de Entidades de Tecnología de Apoyo (AETAP) en colaboración con las empresas de base tecnológica CPTMI S.L. Y CIMA S.L. (Centro de Innovación Multimedia y Animación).

El sistema operativo del iFreeTablet, denominado SieSTA (Sistema Integrado de e-Servicios y Tecnologías de Apoyo) es el resultado de más de 25 años de investigación de alumnos, profesores y técnicos del grupo EATCO de la UCO y de 5 años de colaboración con los integrantes del proyecto UTUTO (distribución Argentina del sistema operativo GNU).



Figura 21: iFreeTablet vista frontal y lateral.

Funcionalidades Clave

- Interacción Persona-PC/TV de forma multimodal y ubicua a través de:
 - Pantalla táctil.
 - Cámara web.
 - Sistema de reconocimiento de voz, movimientos y gestos.
 - Sistema RFID o un mando remoto (Sistema de Interacción Natural, iFreeSIN) que permite a la tablet ser utilizada como dispositivo de entrada de otros dispositivos, como si fuera un control remoto.
- Desarrollo 100% de software libre que incluye:
 - Sistema Operativo libre con kernel Linux.
 - Sistema de televisión por internet.
 - Sistema de radio por internet.
 - Sistema de control domótico de dispositivos digitales del hogar.
 - Sistema comunitario de difusión de eventos por Internet.
 - Accesibilidad dada por un diseño de interfaz basado en la definición de los menús por categorías representadas por botones de colores, de forma que solo estén activos como mucho seis botones.
- Usabilidad dada por interacción multimodal que incluye un barrido multimodal (puede ser usado a través de sonidos como por ejemplo un chasquido de dedos o por toque de la pantalla táctil) a nivel de toda la interfaz que permite manejar el sistema operativo y todas las aplicaciones.



Figura 22: Barrido con entrada multimodal de SieSta.

Especificación Técnica

- Disco duro 160G SATA.
- Conexión Ethernet y puerto wlan WIFI.
- Cámara 1,3 Mpixels.
- Memoria RAM de 1GB (ampliable a 2 GB).
- 3 puertos USB, 1 puerto de salida VGA, 1 conector para auriculares, 1 conector de micrófono, 1 micrófono interno, 1 puerto RJ-45 para LAN, 1 DC-in jack, 1 lector de tarjetas compatible con SD, memory stick, etc.
- 1 ranura SIM.
- 1 ranura DIMM, 2 Mini PCI-E: una para WiFi 802.11b / g 54Mb, otra libre.
- Una batería de 2.5 horas de duración, adaptador de 35W, sistema de refrigeración térmico con ventilador inteligente.
- Dimensiones: 24.6 cm x 16.7 cm y 2.4 cm de grosor, pesa 1 kg (incluyendo la batería).

2.6 OLPC - Plan Ceibal y la Accesibilidad

En la wiki del proyecto OLPC se proponen varias ideas de accesibilidad para distintos tipos de discapacidades. [34] En particular a lo que refiere a la discapacidad motriz propone:

- Hacer OLPC accesible a las personas con discapacidad debe ser uno de los puntos en la declaración de la misión del proyecto. Dado el alto porcentaje de niños con discapacidad en los países en desarrollo, es fundamental que todo el hardware y el software que se agregue a la configuración predeterminada de las computadoras sea de código abierto y de bajo costo.
- Las computadoras deben ser totalmente operables desde el teclado sin necesidad de trackpad para la operación. Algo así como un programa de pintura puede ser una excepción, pero en caso contrario, las tareas como el lanzamiento de programas, navegar por la web o incluso leer un libro (e idealmente la mayoría de los juegos) deben ser totalmente operables desde el teclado. Debe haber una indicación visible de qué objeto tiene el foco del teclado (y esta indicación visual debe ser más evidente en temas de interfaz orientados a personas con problemas de visión)
- Incluir la suite *AccessX* de atajos del teclado.

- Soportar teclados externos y dispositivos de ratones/trackpad.
- Contar con un teclado en la pantalla.
- Agregar un lector de pantalla de código abierto que permita el uso de personas con discapacidad visual.

En lo que refiere a problemas cognitivos, Sugar [35] la interfaz de usuario de las computadoras XO de OLPC, ya aborda muchas de las necesidades de las personas con discapacidades cognitivas. Emplea un uso muy limitado de texto, y proporciona una interfaz despejada. Sin embargo, hay una serie de mejoras adicionales que se pueden realizar para mejorar la interacción de personas con discapacidad:

- Ayudar la comprensión de la lectura a través de definiciones de palabras sensibles al contexto.
- Proporcionar ayuda con la composición de texto a través de la ortografía de búsqueda contextual, y de sinónimos y antónimos sensibles al contexto.
- Utilizar las conversión de texto a voz, de manera de ayudar a la lectura a través de la lectura del texto en voz alta, y además opcionalmente destacando las palabras cómo se leen, sincronizado con el habla.

OLPC marca como tareas a realizar:

- Definir pautas de accesibilidad para el desarrollo tanto como para Sugar como para sus actividades.
- Hacer que el *toolkit* de accesibilidad de *GNOME* [36] [37] funcione en Sugar, con el fin de apoyar las tecnologías de asistencia.

Si bien el Plan Ceibal no define ideas propias sobre la accesibilidad, al utilizar en gran parte la tecnología aportada por OLPC a nivel de equipos y al contar con casi la misma comunidad de desarrolladores en lo que es el desarrollo para Sugar y de actividades, se puede deducir que gran parte de las ideas que propone OLPC de realizarse beneficiarán a la comunidad de usuarios de Plan Ceibal.



Figura 23: Logos OLPC y Plan Ceibal

2.6.1 Equipos

A continuación se describen algunos de los equipos entregados por el Plan Ceibal:

Modelo	Características
 Laptop XO 1.0	CPU: AMD LX700 433 MHZ. RAM: 256 Mbyte DDR SDRAM, bus de 333 MHZ. Almacenamiento: 1 GB de memoria flash NAND, 1 MB de memoria flash serial utilizada para el firmware, posibilidad de expandir la memoria de almacenamiento a través de un socket para memorias del tipo SD/MMC. Sistema operativo: Distribución Fedora 14.0 (Kernel 2.6.35.13) Interfaz de usuario: Sugar 0.94.1 (Dextrose 3)
 Laptop XO 1.5	CPU: VIA C7-M 1 Ghz. RAM: 1 Gbyte DDR2 SDRAM, bus de 400 MHZ. Almacenamiento: 8 GB de memoria flash NAND (microSD en la motherboard), 1 MB para la BIOS, flash ROM, posibilidad de expandir la memoria de almacenamiento a través de un socket para memorias del tipo SD/MMC. Sistema operativo: Distribución Fedora 14.0. Interfaz de usuario: Sugar 0.94.1 - dextrose 3 y Gnome 2.32.0.
 Laptop XO 4.0	CPU: dual Marvell PJ4 ARM cores @1GHz (ARMv7l). RAM: 1 Gbyte DDR3. Almacenamiento: 8 GB de memoria flash eMMC, 1 MB de memoria flash serial utilizada para el firmware, posibilidad de expandir la memoria de almacenamiento a través de un socket para memorias del tipo SD/MMC. Sistema operativo: Distribución Fedora 18 Interfaz de usuario: Sugar 0.98.7 - Dextrose y Gnome 3.6.21
 Laptop XO 4.0 V2	CPU: dual Marvell PJ4 ARM cores @1GHz (ARMv7l) RAM: 2 Gbyte DDR3 Almacenamiento: 16 GB de memoria flash eMMC, 1 MB de memoria flash serial utilizada para el firmware, posibilidad de expandir la memoria de almacenamiento a través de un socket para memorias del tipo SD/MMC. Sistema operativo: Fedora 20 y Android 4.3 Interfaz de usuario: Fedora: Sugar 0.104 - Dextrose y Mate 1.8.1 Android: Android 4.3.1

 Olidata JumPC/FK1	CPU: Intel Celeron 900 Mhz. RAM: 1 Gbyte. Controlador de video integrado en Chipset Intel Mobile 915GM. Almacenamiento: 8 Gb SSD o disco de 30Gb. Sistema operativo: Ubuntu 12.04 (Kernel 3.8.0-33-generic) Interfaz de usuario: Gnome y Sugar.
 Magallanes MG1	CPU: Intel Atom N270 1,66 Ghz. RAM: 1 Gbyte DDR2. Almacenamiento: 8 Gb SSD o disco de 30Gb. Sistema operativo: Ubuntu 12.04 (Kernel 3.8.0-33-generic) y Windows XP (con una versión para no videntes) Interfaz de usuario: Gnome y Sugar.
 Magallanes MG2	CPU: Intel Atom N450 1,66 Ghz. RAM: 1 Gbyte DDR2. Almacenamiento: Versión SSD: Capacidad: 8 GB, posibilidad de expandir la memoria de almacenamiento a través de un socket para memorias del tipo SD/MMC. Versión HDD: Capacidad: 160 GB o 320 GB. Sistema operativo: Ubuntu 12.04 (Kernel 3.8.0-33-generic) y Windows XP (con una versión para invidentes) Interfaz de usuario: Gnome y Sugar.
 XO Tablet	CPU: Rockchip RK3066, dual core 1.6GHz. RAM: 1 Gbyte. Almacenamiento: 8 GB NAND Flash. Sistema operativo: Android 4.1.1 (Jelly Bean).

 Tablet JP	CPU: Intel Atom Z2520 1.2GHz x 2 (Dual Core). RAM: 1 Gbyte Almacenamiento: 8 GB eMMC Sistema operativo: Android 4.2.2 (Jelly Bean)
---	--

2.6.2 Actividades

A continuación se presentan algunas funcionalidades y actividades de XO/Sugar que están orientadas a la accesibilidad.

2.6.2.1 Hablar/Hablar con Alicia

Hablar es una actividad que le da una cara parlante a Sugar con un área para escribir, donde todo lo que se escriba reproducido en modalidad auditiva utilizando el sintetizador de voz “*espeak*”. Permite realizar ajustes del acento, velocidades y al timbre de la voz, y personalizar la cara cambiando la forma de la boca y los ojos.

Hablar con Alicia es una versión modificada de *Hablar* que embebe inteligencia artificial a través de *PyAIML* [38] y un *chatbot* [39] .



Figura 24: Actividad **Hablando con Alicia**

2.6.2.2 Escribir Especial

Consiste en un teclado en pantalla que permite escribir con un solo botón por medio de técnicas de

barrido dentro de un cuadro que aparece en la parte inferior de la pantalla.

Se puede escribir en el cuadro de diferentes maneras:

- Con el teclado de la XO (como se escribe normalmente)
- Con el teclado que aparece en pantalla: apretando sobre la tecla con el puntero del ratón mediante la opción de barrido.

Para el barrido se pueden elegir 3 velocidades: rápido, medio o lento.

Esta Actividad dejó de mantenerse y funciona para las versiones posteriores de a la XO 1.75.



Figura 25: Actividad **Escribir**

2.6.2.3 PUI: *Perceptual User Interface*

XO-PUI [40] es un framework para la captura de gestos y/o formas utilizando la cámara incorporada a las XO, y la generación de un evento a partir de dicha captura. Fue desarrollado como proyecto de grado en la Facultad de Ingeniería de la UdelAR por los ingenieros Marcela Bonilla y Sebastián Marichal en el año 2011.



Figura 26: Niña interactuando con cuento interactivo por medio de carteles con imágenes reconocidas utilizando PUI.

Capítulo 3

Contexto del Proyecto

El contexto en donde se realizó el proyecto es el de la Escuela N° 200 Dr Ricardo Caritat, escuela pública de Montevideo para niños y niñas con discapacidad motriz. Funciona desde 1978 y tiene como objetivo habilitar y/o rehabilitar a los alumnos para que desarrollen al máximo sus potencialidades y así mejorar su inclusión. Este objetivo se persigue empleando diferentes estrategias y herramientas, entre ellas la informática.



Figura 27: Escuela N° 200 Ricardo Caritat

La institución brinda servicios educativos en los niveles de jardinera y primaria, siguiendo el Programa de Educación Inicial y Primaria de las escuelas comunes adaptándolo a las necesidades de cada chico. La mayoría, son niños en edad escolar aunque también asisten algunos alumnos adolescentes, abarcando así un rango de edades entre los dos y los veinte años.

El plantel docente es integrado por la directora, diez maestras, dos fisioterapeutas, una profesora de computación, un musicoterapeuta, una psicóloga, una trabajadora social y nueve asistentes entre otros empleados. Las maestras trabajan todo el año lectivo con el mismo grupo, sin embargo, es criterio adoptado por la institución que las maestras no mantengan por más de dos años consecutivos a los mismos niños con el fin de prevenir un deterioro en las relaciones con las familias de los chicos.

Los alumnos asisten a la escuela desde todos los barrios del departamento de Montevideo, el 80% de ellos viven en un contexto social crítico. La mayoría de los niños concurre 3 horas al día en la escuela a excepción de un grupo que lo hace durante 6 horas.

El ingreso se produce luego de que se evalúa el niño por medio de una entrevista y habiendo cupo en el grupo al cual les corresponde asistir. El proceso de egreso se da cuando el niño culmina la rehabilitación o habilitación y se busca la institución adecuada en donde se pueda inscribir.

Los perfiles de admisión son variados, ingresan desde niños con patologías muy leves que estando incluidos en escuelas comunes reciben en Caritat apoyo pedagógico hasta niños con patologías muy

severas.

Al centro concurren 47 niñas y 55 niños con diferente tipo de discapacidad motriz: 90% tiene Parálisis Cerebral, 8% tiene Espina Bífida y un 2% presenta otras patologías.

A continuación se describen las características generales que presentan la mayoría de los niños que asisten a la escuela con el fin de mostrar las dificultades de llevar a cabo un proceso de diseño de interfaces de usuario en este contexto particular.

3.1 Patologías de los niños

La gran mayoría de los niños que concurren a la escuela tienen Parálisis Cerebral y dentro de sus patologías encontramos las que se mencionaron en la sección sobre PC en el capítulo 2, estas incluyen:

- **Parálisis Cerebral Espástica**
- **Parálisis Cerebral Discinética**
- **Parálisis Cerebral Atáxica**

Además de las limitaciones físicas y cognitivas esperables según la categoría, , relevamos en los niños de la escuela las siguientes particularidades:

3.1.1 Características del aprendizaje

Para que un niño incorpore símbolos, ya sean letras o números, es fundamental que sepa discriminar.

Esto implica:

- Reconocimiento de las cualidades de un objeto a través de los sentidos - SENSACIÓN.
- Reconocimiento inmediato del objeto aunque el estímulo que lo provoque no sea completo - PERCEPCIÓN. Es aquí donde se suman las diferentes experiencias que posee un individuo.

Para que un niño incorpore símbolos, ya sean letras o números, es fundamental que sepa discriminar. Esta habilidad implica por un lado el reconocimiento de las cualidades de un objeto a través de los sentidos, lo que llamamos sensación, y por otro el reconocimiento inmediato del objeto aunque el estímulo que lo provoque no sea completo, es decir, la percepción. En este último proceso es donde se integran las diferentes experiencias que posee un individuo. Además, la percepción también es fundamental para organizar y estructurar el espacio, e integrar, organizar e interpretar elementos sonoros. Encontramos que los niños de la escuela presentan déficits para realizar estas funciones. En particular, las siguientes características parecen ser responsables de las dificultades para discriminar, a saber:

Hiperactividad: Responsable de la incapacidad para mantener la atención, de una conducta distraída a nivel visual y auditivo, y de las perturbaciones en la percepción.

Distractibilidad: Incapacidad del niño para centrar en forma adecuada la atención sobre los estímulos que le interesan e inhibir voluntariamente los innecesarios.

Disociación: Incapacidad para ver las cosas como un todo, captando antes las partes (dibuja, arma sin integrar una figura, lo hace en forma desorganizada).

Perturbación de figura-fondo: Tendencia a confundir la figura y el fondo, invertir fondo y figura o incapacidad de diferenciar la figura del fondo.

Perseverancia: Tendencia a continuar una actividad sin que se complete.

Alteraciones de la imagen corporal y del concepto de sí mismo: Al no elaborar una buena imagen corporal, su propio auto concepto se verá deteriorado y los aprendizajes resultan perturbados.

Inestabilidad en el rendimiento: Inconstancia en las respuestas de aprendizaje, fluctuando su rendimiento de un día a otro aparentando "haber perdido" conocimientos adquiridos el día anterior, "olvidos" originados por fallas en la interiorización.

Discordancia en los movimientos: Diferencias notorias entre las habilidades motoras finas y gruesas.

Deficiencias en las áreas funcionales:

- Dificultad para recepcionar consistentemente la información del medio ambiente.
- Dificultades en el procesamiento de la información.
- Cierre rápido, apropiado y correcto que le permita dar una respuesta esperada.

3.1.2 Comunicación

Uno de los objetivos principales de la escuela es de desarrollar un sistema de comunicación y así dignificar la comprensión del niño para mejorar su calidad de vida.

Los procedimientos, recursos y materiales que se emplean son tan diversos como las heterogeneidad de casos a los que se enfrentan.

Comunicación oral y gestual

En ausencia de habla:

a) Buscar siempre mejorar su postura (de cuerpo y cabeza) para que el niño:

- normalice su mímica facial.
- mantenga boca cerrada y lengua quieta.
- respire por la nariz.
- babee menos (sialorrea).
- mejore deglución y masticación.

b) Investigar el tiempo que necesita cada niño para comprender un mensaje y responder, lo que implica para él un gran esfuerzo y concentración.

c) Explorar posibilidades expresivas tales como:

- recorridos visuales
- movimientos de cabeza
- señalización

d) Elaborar un menú o código de respuestas de acuerdo a las posibilidades exploradas.

Comunicación Escrita

Como actividad neuro-psico-motora involucra:

- Coordinación de músculos, hombros, brazo, columna, movimientos de muñeca, trípode estático y dinámico.
- Ejercicios de prensión y trazado.
- Ejercicios de destrezas perceptivo-motoras y de integración sensoriomotora.
- Tener definida la mano que predomina en caso de no ser una dominancia forzada por la discapacidad que presenta.

Diferentes casos

Presión leve

El niño puede sujetar con el puño, dedos cruzados o movimientos de pinza el útil, pero se le cae muchísimas veces. Se necesita: fijar el útil a su mano dominante (velcro, cinta, pulsera con tiras cruzadas, útil imantado a la pulsera). Esto evitará pérdida de tiempo y frustración y mejorará así su capacidad de concentración.

Presión excesiva o exagerada

El niño sujeta bien el útil pero ejerce una presión exagerada rompiendo el lápiz y hojas. Se necesita usar un mango resistente que forre al lápiz, pincel o drypen (forro de caucho, metal o plástico). Esto los hace más gruesos, fáciles de asir y resistentes. Además estos útiles deben estar siempre cerca de ellos y fácil de asir sin ayuda (soporte). Para evitar que la hoja se rompa y resbale, necesita una hoja dura de cartón o cartulina pegada a la mesa (cinta, chinchas, elástico, etc.).

Movimientos involuntarios y bruscos (Espasmos)

Se usan muñequeras con peso indicado por el fisioterapeuta durante el tiempo que necesite concentrarse en una actividad motriz

Cuadripléjico severo con control de movimiento de cabeza relativamente bueno.

Puede atarse un puntero o regla a un casco, o emplear el útil en la boca (ambos necesitan un buen soporte). Se sugiere un tipo de letra de fácil trazado: imprenta mayúscula. En caso de no poder lograr el movimiento de trazado se lo habilitará para el uso de la máquina de escribir con teclado especial y/o computadora.

Otros materiales a utilizar son los sellos con moldes de letras o los cubos de madera con letras impresas. El tamaño de la hoja, renglón, letra y distancia entre las líneas podrán reducirse gradualmente según los distintos casos.

Se debe tratar siempre la no inutilización del otro brazo, por ejemplo puede apoyarlo sobre la hoja para sujetarla o asir un soporte para evitar que esconda o cuelgue el brazo.

3.1.3 Recursos Didácticos y Consideraciones Generales

Los materiales y recursos que se utilizan en la escuela son los que emplea todo maestro pero:

- Utilizan el mayor número de sentidos posibles.
- Reafirman con mayor repetición.
- Siguen un ritmo lento pero continuo.
- Estimulan pero no sobreestiman.
- Evitan fracasos y frustraciones.

Además, las deficiencias en las áreas funcionales han de reducirse y controlarse :

- Eliminando información no relevante, redundante, distrayente y ambigua.
- Programando y presentando la información que tenga valor, que le permita dar respuestas efectivas.
- Automatizando los procesos de aprendizaje para que el niño no pierda continuidad.

También han de seleccionarse las tareas que minimicen:

- Distractibilidad: tareas con pocos detalles y que finalicen rápidamente, poco estímulo ambiental.
- Desinhibición motora: evitar la movilidad y el constante cambio postural que disminuyan sus posibilidades de atención.
- Perseverancia: variar las tareas, que no se parezcan en forma y modalidad.
- Fallas en la memoria: tareas de pasos breves.
- Falta de voluntad. Reforzar la motivación y la responsabilidad
- Lentitud: tareas breves y demostrar paciencia para su modalidad lenta.

Capítulo 4

Problemática detectada y Descripción de la Solución

En este capítulo se presenta la problemática puntual a resolver por el proyecto y una descripción de la solución propuesta. Tanto la problemática como la solución son productos de un proceso de análisis y diseño, siendo el relevamiento de la problemática la primer actividad en la cual se basan todas las siguientes.

A continuación se da una descripción el proceso realizado la cual incluye el problema y la solución.

4.1 Proceso de Análisis y Diseño

El proceso de análisis y diseño sigue modelo de proceso de análisis y desarrollo tradicional, pero también toma ideas del modelo *MPlu+a* presentado en el capítulo 2 de manera de centrar el diseño en el usuario y realizar una evaluación incremental de usabilidad y accesibilidad.

También se consideran los siete principios del diseño universal y las pautas de diseño orientado a niños, en particular a niños con discapacidad. Esta mezcla de ideas y pautas confluye en un proceso de análisis y diseño iterativo e incremental en el que hay una evaluación continua por parte de los usuarios. A continuación se describen las actividades que se realizaron durante el proceso.

4.1.1 Actividad 1: Análisis de Requisitos

El análisis de requisitos consistió de la realización de varias actividades, la gran mayoría de ellas se realizaron durante la primer visita a la escuela, la cual se hizo luego de haber reunido la información disponible sobre la escuela en internet, de manera de contar con una base de conocimiento sobre la realidad con la que se iba a trabajar.

Primer visita a la Escuela

Esta visita primer visita se consistió en una recorrida por toda la escuela, pasando por todos los salones de clase y conociendo a los niños y las maestras. La problemática a resolver es la relevada a partir de las diferentes actividades realizadas durante ésta recorrida.

4.1.1.1 Análisis Etnográfico

Durante la recorrida se realizó un análisis etnográfico de los niños y las maestras, mediante las observación de cómo se realizaban parte de las clases y en algunos casos como la interacción de los niños con las computadoras que el Plan Ceibal les entregó.

A partir de este análisis se tomaron las siguiente notas:

- La comunicación entre las maestras y sus alumnos es la base de la educación en la escuela. Se observó que en los casos en donde la capacidad de comunicación del niño es muy limitada o nula, es necesario a utilizar herramientas como comunicadores con imágenes. A través de su empleo y con la asistencia de las maestras, los niños pueden realizar una comunicación muy básica.



Figura 28: Cuaderno Comunicador

- No cuentan en la escuela con una aplicación equivalente a un comunicador. Esto implica que haya que armar uno a uno los comunicadores personalizados en papel que suelen romperse y a veces se pierden cuando los llevan a la casa. Este hecho denota la necesidad de contar con una herramienta que pueda utilizarse con ese fin.
- La gran mayoría de los casos en que los niños tenían una capacidad de comunicación muy limitada o nula, se corresponden con los casos de discapacidad motriz más severos, como por ejemplo niños con tetraplejia. Para estos casos en la escuela se intenta que pese a ello, utilicen sus computadoras, a veces se logra con la ayuda de pulsadores caseros y en otras manejando las computadoras con los pies, pero en todos los casos la interacción requiere de mucho esfuerzo al no contarse con aplicaciones como un teclado en pantalla con barrido como el SwitchXS.
- Si bien una aplicación como esta, que permita a los niños manejar todas las funcionalidades de la computadora a través de un solo evento, lograría prácticamente solucionar sus problemas de comunicación, exige un nivel cognitivo muy elevado debido a su complejidad. A saber, contar con muchos elementos, presentar solamente texto y realizar una gran cantidad de acciones diversas. Teniendo en cuenta que el desarrollo del nivel cognitivo del niño está más comprometido cuanto más severa es su discapacidad motriz, una aplicación con un diseño similar al SwitchXS sería poco usable para los que más se beneficiarían de su uso.

4.1.1.2 Plataforma

También se relevó que en su mayoría utilizan computadoras de modelos Magallanes con

Ubuntu/Gnome como sistema operativo e interfaz de usuario respectivamente y en algunos casos también se utiliza una tablet con sistema operativo Android.

4.1.1.3 Objetivos de Usabilidad

Los objetivos de usabilidad detectados son:

- sea usable por niños con una discapacidad motriz severa
- sea usable por niños con un bajo nivel cognitivo

Luego de la primer visita a la escuela, habiendo relevado la problemática a resolver se pasó a definir qué herramienta construir y describir sus requerimientos.

4.1.1.4 Descripción de Requerimientos

A partir de lo problemática detectada relevada y en la información obtenida en las actividades anterior se decide como solución a la problemática analizada, desarrollar un **teclado en pantalla** que cumpla los siguientes requerimientos:

4.1.1.4.1 Requerimientos Funcionales

1. Contar con teclas que presenten texto y/o imagen.
2. Que las teclas puedan emitir sonidos y realizar acciones, como por ejemplo abrir el navegador, mover el puntero del mouse o apagar la computadora.
3. Sea totalmente personalizable pudiendo escoger las teclas que aparecen en pantalla, su contenido (texto, imagen y sonido) y el orden de las mismas.
4. Pueda ser manejado con un solo botón utilizando la técnica de barrido, pudiendo configurar el barrido agrupando las teclas a gusto y la velocidad del barrido.
5. Permita crear configuraciones personalizadas para distintos usuarios, modificar estas configuraciones y así aumentar o disminuir la complejidad de la aplicación.

Los puntos 1 y 2 siguen el cuarto principio de diseño universal al comunicarse la información de modo multimodal de manera de atender las condiciones ambientales o a las capacidades sensoriales del usuario.

El punto 3 sigue el tercer principio de diseño universal, brindando la posibilidad de personalizar los elementos para que el contenido sea fácil de entender, atienda a la experiencia, conocimientos, habilidades lingüísticas o grado de concentración actual del usuario.

El punto 4 sigue el sexto principio del diseño universal al exigir poco esfuerzo físico minimizando las fuerzas necesarias para operar.

El punto 5 sigue el segundo principio de diseño universal que propone un uso flexible que ofrezca posibilidades de elección en los métodos de uso, se confeccionan unos requerimientos que contemplen una evolución cognitiva de los usuarios, esto implica que la herramienta también

evolucione a la par de los usuarios.

4.1.1.4.2 Requerimientos no Funcionales

1. Tenga un diseño que contemple los principios del diseño universal y de diseño orientado a niños con discapacidad.
2. Funcione en las computadoras (modelo Magallanes) de los niños de la escuela y en tablets Android dado que se prevé que se puedan utilizar en la escuela en un tiempo próximo.

4.1.2 Actividad 2: Diseño

En el paso siguiente se confeccionó una descripción de las funcionalidades, con el objetivo de tener claro qué debería cumplir la aplicación a nivel funcional.

4.1.2.1 Funcionalidades

Alta y Modificación de Teclas

La base del teclado son las teclas. Por este motivo se deben poder crear y modificar las teclas que se usan para armar lo distintos modos del teclado. Tanto en la creación como en la modificación un usuario debe poder:

1. Ingresar un texto para la tecla, el cual va a ser el texto que la identifique.
2. Seleccionar una imagen representativa para la tecla, que se va a seleccionar entre las imágenes que se encuentran en la computadora o tablet del usuario.
3. Seleccionar un sonido que se reproduzca cuando se presione la tecla, que se va a seleccionar de los sonidos que se encuentran dentro de la computadora o tablet del usuario.
4. Asignar una categoría a la tecla de modo que sea sencillo encontrarla luego para agregarla a un modo.
5. Seleccionar un script para que se ejecute al presionar la tecla, el mismo debe estar en la computadora o tablet del usuario con los permisos de ejecución necesarios.

Alta y Modificación de Modo

Los modos son los diferentes *layouts* que están ingresados en la aplicación. Cada modo consta de:

- Las teclas que lo componen.
- Una configuración propia del modo.
- Un barrido en el que se define el cómo el escáner hará el barrido las teclas.

Tanto al crear o modificar un modo un usuario debe poder:

1. Agregar y quitar las teclas que componen un modo.
2. Configurar el modo.
3. Definir y modificar la definición del barrido.

Configurar Modo

La configuración del modo permite a los usuarios adaptarlo. adaptarlo para lograr esa evolución a medida que cambien las necesidades del usuario o demande un contenido de mayor complejidad.

Esta versatilidad también hace que sea una herramienta que se ajusta a una gran variedad de perfiles cognitivos. Para ello se consideraron los siguiente parámetros para poder modificar:

- Si las teclas presenta el texto o no.
- Si las teclas presenta una imagen o no.
- Si las teclas reproduce un sonido o no.
- Si las teclas envían teclas al sistema operativo (teclado virtual clásico).
- Si las teclas ejecutan scripts o no.
- Velocidad del Barrido: la velocidad con que el barrido cambia la selección.
- Cantidad máxima de teclas que se muestran.

Se deben poder modificar estos parámetros para cada modo que se encuentre ingresado.

Seleccionar Modo

Se debe poder cambiar el modo que se está utilizado en cualquier momento.

Configurar Barrido de Modo

Dadas las teclas de un modo, se debe poder definir un barrido particular del modo en que se definan:

- Agrupaciones de teclas que permitan seleccionar más rápido la tecla deseada.
- A su vez estos grupos pueden estar dentro de otro grupo para acelerar aún más el proceso.
- El orden en que se barren estos grupos.
- La velocidad con que se barre (esto se realiza en la configuración de modo)

Iniciar y Detener Barrido

Se debe poder iniciar y detener el barrido.

4.1.2.2 Diseño Detallado

A partir de la descripción de las funcionalidades, se pasó a la realización del diseño de las interfaz de usuario. Siguiendo el proceso primero se realizaron varios modelos conceptuales en papel los cuales fueron generando un diseño detallado.

Para la definición del estilo se utilizaron pautas del diseño de interacción orientado a niños (ver capítulo 2) como no utilizar tamaños de fuente pequeños y usar un buen contraste de color, y el séptimo principio de diseño universal proporcionando un tamaño y espacio apropiados para el acceso, alcance, manipulación y uso, atendiendo al tamaño del cuerpo, la postura o la movilidad del usuario, por medio del uso de elementos grandes y diagramación de pantallas simples con pocas opciones.

Como herramienta para generar los diseños se utilizo *Pencil* [41] de código abierto, muy útil para realizar prototipos de interfaz gráfica debido a que cuenta con los elementos tradicionales para armar las pantallas y también con versiones de estilos estándar de elementos, por ejemplo elementos para Android.

A continuación se presentan las pantallas y la navegación entre las mismas.

4.1.2.2.1 Pantallas

Pantalla Principal

En la pantalla principal es donde se presenta el modo, está compuesta por dos partes:

- A la izquierda se encuentra el teclado con el modo seleccionado.
 - Qué presentan las teclas dependiendo de la configuración del modo, pudiendo presentar texto y/o imagen.
 - El tamaño de las teclas depende de la cantidad de teclas que tenga el modo.
- A la derecha el panel de opciones y la información de estado. Dentro del panel de opciones se tiene:
 - La activación y desactivación del barrido.
 - La selección del modo a presentar. Abre la selección de modos.
 - El nombre del modo seleccionado.
 - La opción de ir a la edición del modo seleccionado. Abre el editor de modos.
 - La opción de configurar el modo seleccionado. Abre la configuración de modos.
 - La opción de salir de la aplicación.



Figura 29: Pantalla Principal, el modo presenta un teclado qwerty.



Figura 30: Pantalla Principal, teclas con solo texto. Este modo contiene palabras armadas: pronombres y artículos.

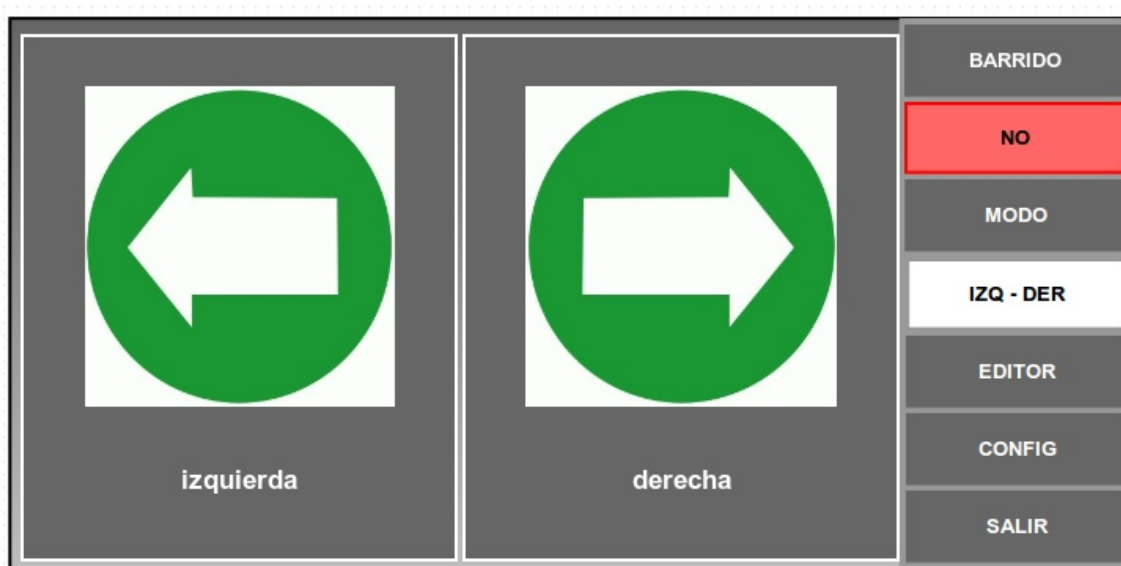


Figura 31: Pantalla Principal con un modo cargado con teclas con texto e imagen. Este modo podría ser útil para enseñar cuál es la izquierda y cuál la derecha.

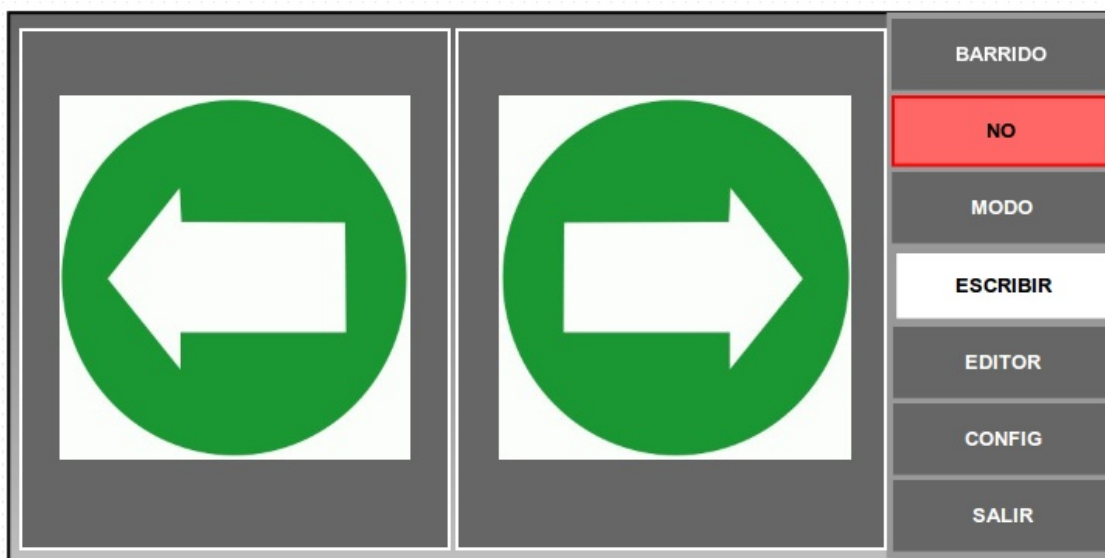


Figura 32: Pantalla Principal con el mismo modo que la figura anterior pero configurado para que solo muestre las imágenes.

Barrido Activado

Al activarse el barrido se comienza a recorrer las teclas según el barrido definido para el modo. Para activarlo o desactivarlo hay que presionar el botón de barrido en el panel de opciones a la derecha, quedando indicado el estado de activo en el menú.



Figura 33: Barrido activado

Editor

El editor es donde se crean y modifican los modos existentes. También está dividido en dos partes, a la izquierda un panel donde se van a ir acumulando las teclas que se se vayan agregando y a la derecha las opciones:

- **Agregar Tecla:** Abre la pantalla para agregar teclas, pudiendo elegir allí agregar una tecla ya existente o crear una nueva.
- **Editar Tecla:** Habiendo seleccionado una tecla de las agregadas permite editar el contenido de la tecla. Es decir texto, imagen, sonido y acción.
- **Quitar Tecla:** Habiendo seleccionado una tecla la quita del modo.
- **Quitar Todo:** Quita todas las teclas del modo, dejando el modo vacío.
- **Definir Barrido:** Abre la pantalla que permite definir el barrido para el modo a partir de las teclas del mismo.
- **Guardar:** Guarda los cambios realizados.
- **Salir:** Vuelve a la pantalla principal.

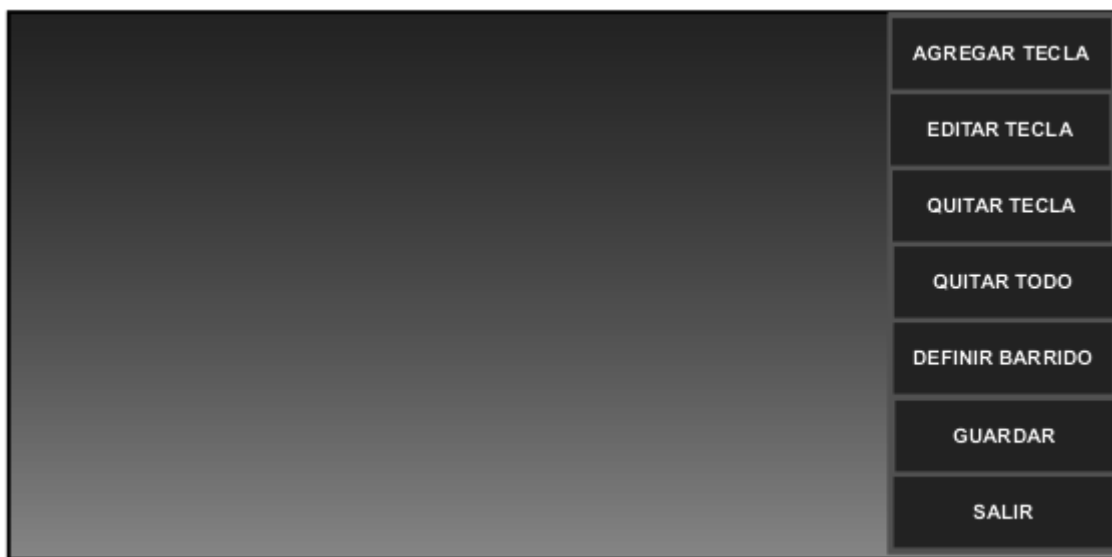


Figura 34: Pantalla de Editor de Modos.



Figura 35: Editor al que se le agregó una tecla con el texto “1” sin imagen.

Agregar Tecla

Tiene la misma diagramación que las otras, a la izquierda teclas y a la derecha un menú de opciones:

1. **Categorías:** Las primeras opciones representan las categorías de las teclas. Cuando se selecciona una categoría, se cargan a la izquierda las teclas que contiene como forma de seleccionar una.
2. **Nueva Tecla:** En caso de que no exista la tecla que se quiere agregar se da la opción de crear una nueva tecla. Abre la pantalla de creación y modificación de teclas.

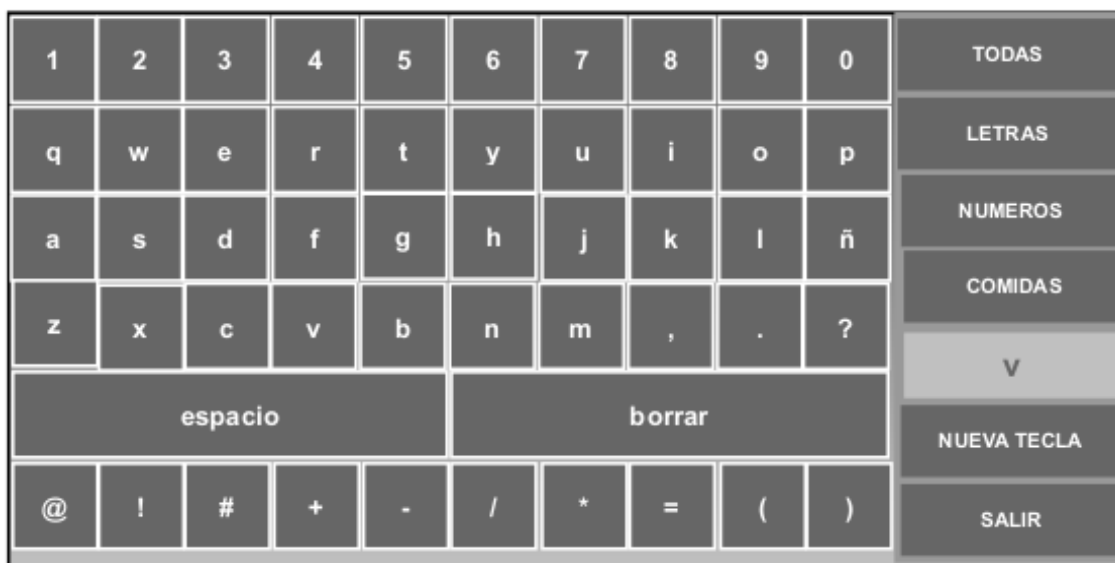


Figura 36: Categoría cargada con las teclas del qwerty. (La v en el botón del panel de opciones no es un v sino una flecha que implica que se presentan más opciones cuando se presiona el botón, corriendo las opciones que están arriba del botón)

Definir Barrido

Una vez que se tienen definidas las teclas que componen un modo se puede definir el modo en que el barrido las recorre, para ello se utiliza la pantalla de definir barrido.

La pantalla cuenta con dos partes, a la izquierda las teclas que componen el modo y a la derecha el menú de opciones:

- **Niveles:** La idea del barrido es poder seleccionar una tecla de forma asistida. El barrido va recorriendo las teclas de forma individual o de a grupos de teclas. Si el usuario dispara el evento de selección estando el barrido sobre una tecla se realizan las acciones como si se apretara la tecla. Si el barrido está sobre un grupo de teclas comienza el barrido sobre el contenido del grupo. Para definir estos grupos se utilizan los niveles. Un nivel marca la profundidad a la que se encuentra un grupo. Para agrupar teclas para un nivel se dibuja un rectángulo que cubra las teclas a agrupar habiendo seleccionado el nivel en el menú de opciones. Para definir un subgrupo se debe seleccionar el nivel siguiente y dibujar un nuevo

rectángulo que agrupe el nuevo grupo de teclas. El barrido se realiza por nivel y en el orden en que fueron dibujados los rectángulos. Para agregar un nuevo nivel se presiona **Nuevo Nivel**.

- **Guardar:** Guarda el barrido definido para el modo.
- **Salir:** Vuelve al Editor de Modos.

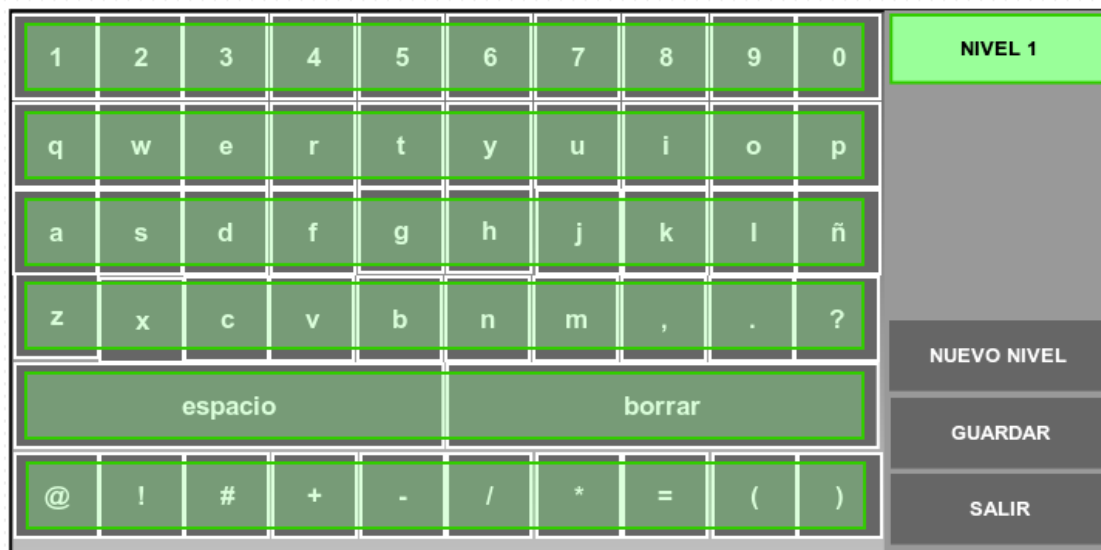


Figura 37: Agrupación de teclas sobre el modo escribir para un nivel 1.

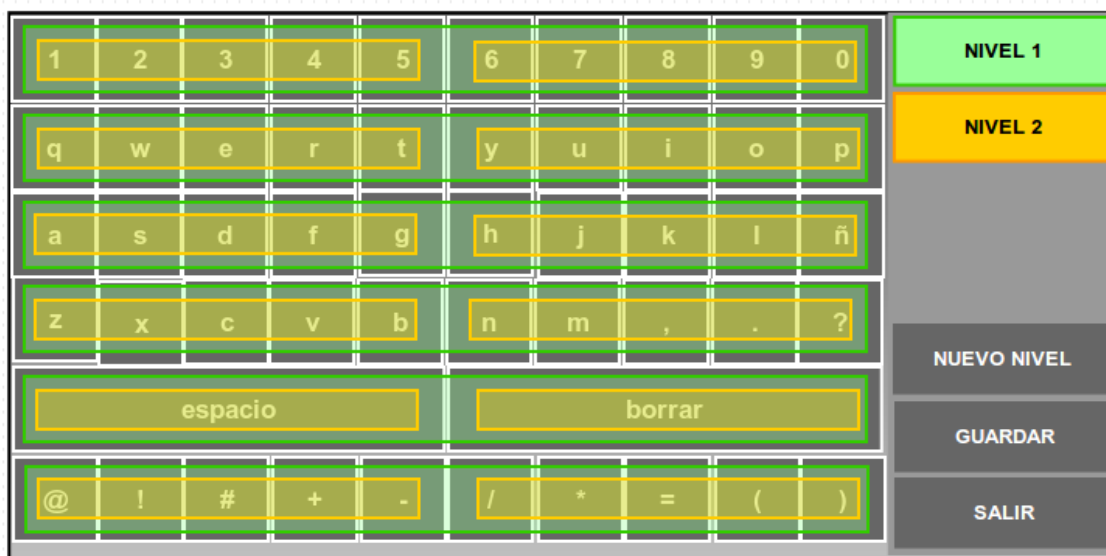


Figura 38: Agrupación de teclas para el modo de escribir en un nivel 2.

El barrido definido en las figuras anteriores se haría de la siguiente manera:

Primero el nivel 1:

- [1,2,3,4,5,6,7,8,9,0]

- [q,w,e,r,t,y,u,i,o,p]
- [a,s,d,f,g,h,j,k,l,ñ]
- [z,x,c,v,b,n,m,,,.?]
- [espacio, borrar]
- [@,!,#,+,-,/,*=,(,)]

Si no se dispara el evento de selección queda barriendo estos grupos. Si por ejemplo se dispara el evento estando en [1,2,3,4,5,6,7,8,9,0] pasa a barrer el grupo y como tiene subgrupos pasa a barrer:

- [1,2,3,4,5]
- [6,7,8,9,0]

Si no se dispara el evento de selección y terminó el barrido de los subgrupos, comienza de nuevo a barrer el nivel 1. Si por ejemplo se dispara el evento estando en [6,7,8,9,0] pasa a barrer ese grupo, como no tiene subgrupos barre directamente las teclas:

- 6
- 7
- 8
- 9
- 0

De nuevo, si no se dispara el evento de selección antes que termine de barrer comienza el barrido desde el nivel 1. Si por ejemplo se dispara el evento estando sobre la tecla “6” se:

- Reproduce el sonido si la tecla tiene un sonido asociado.
- Se realiza la acción si la tecla tiene una acción asociada.
- Se envía el texto a otra aplicación que esté ejecutándose detrás del teclado en caso de haber una.

Crear y Modificar Tecla

La pantalla de creación y modificación de teclas es donde se crean los elementos básicos de los modos. Permite ingresar y modificar los siguientes elementos que componen una tecla:

- **Nombre:** texto que identifica a la tecla, es también el texto que se envía a otra aplicación que tenga el foco de entrada de texto.
- **Imagen:** es la imagen que muestra la tecla, es cargada del sistema de archivos de la computadora o tablet en la que está corriendo la aplicación, por lo que al hacer presionar sobre el cuadrado donde va la imagen se abre el navegador del sistema de archivos para seleccionar la imagen.
- **Sonido:** es el sonido que se reproduce al presionar la tecla, es cargado del sistema de archivos por lo que al presionar para ingresar el sonido se abre el navegador del sistema de archivos para poder seleccionar el sonido a reproducir.
- **Categoría:** es la categoría a la que pertenece la tecla, una tecla puede pertenecer a una sola categoría, puede asignarse una categoría existente o una nueva.
- **Acción:** es el script que se ejecuta cuando se presiona una tecla, es cargado del sistema de

archivos porque al presionar para ingresar el script se abre el navegador del sistema de archivos para poder seleccionar el script a ejecutar.



✓ IMAGEN

✓ NOMBRE PANCHOS

✓ SONIDO panchos.ogg

✓ CATEGORIA COMIDA

✓ ACCION NINGUNA

VISTA PREVIA GUARDAR SALIR

Figura 39: Pantalla de Creación de Teclas

Configuración de Modo

En la configuración de modo es donde se ajustan los parámetros que ajustan la aplicación al nivel cognitivo actual del usuario y a la actividad que se quiera realizar con ella. Los parámetros que se pueden modificar son:

- Máxima cantidad de teclas
- Velocidad del barrido
- Si las teclas muestran texto o no
- Si las teclas muestran imagen o no
- Si las teclas reproducen un sonido o no
- Si las teclas ejecutan una acción o no

CONFIGURACIONES ALVARITO 1 ▼

MAX. CANTIDAD DE TECLAS 4 ▼

VELOCIDAD DEL BARRIDO 1 ▼

USA TECLAS CON:

- ☒ TEXTO
- ☒ IMAGEN
- ☒ SONIDO
- ☐ ACCION

ACEPTAR GUARDAR SALIR

Figura 40: Pantalla de configuración para el modo ALVARITO 1

En la figura se muestra a modo de ejemplo una configuración en la que el modo tiene hasta 4 teclas, con una velocidad de barrido baja = 1, las teclas presentan texto e imagen y reproducen sonido pero no ejecutan acciones. Este sería un modo para un nivel cognitivo más bien bajo.

Selección de Modo

La pantalla de selección de modo permite la seleccionar un modo de entre los modos existentes para cargar en la pantalla principal. Al cargar el modo no solo se cargan las teclas sino que también se carga la configuración del modo y el barrido que tiene definido, si no tiene el barrido definido se hace tecla por tecla.

La pantalla presenta botones con los nombres de los modos de manera que el usuario presione el modo que desea seleccionar. Al presionarlo se presenta la pantalla principal con el modo seleccionado cargado. Si se presiona salir se pasa a la pantalla principal sin cambiar de modo.

ESCRIBIR	PINTAR	COMIDAS	PERSONAS	IZQ/DER	MODO
MODO	MODO	MODO	MODO	MODO	MODO
MODO	MODO	MODO	MODO	MODO	MODO

SALIR

Figura 41: Pantalla de Selección de Modo

4.1.2.2 Navegación

A continuación se presenta un diagrama con la navegación entre las pantallas definidas.

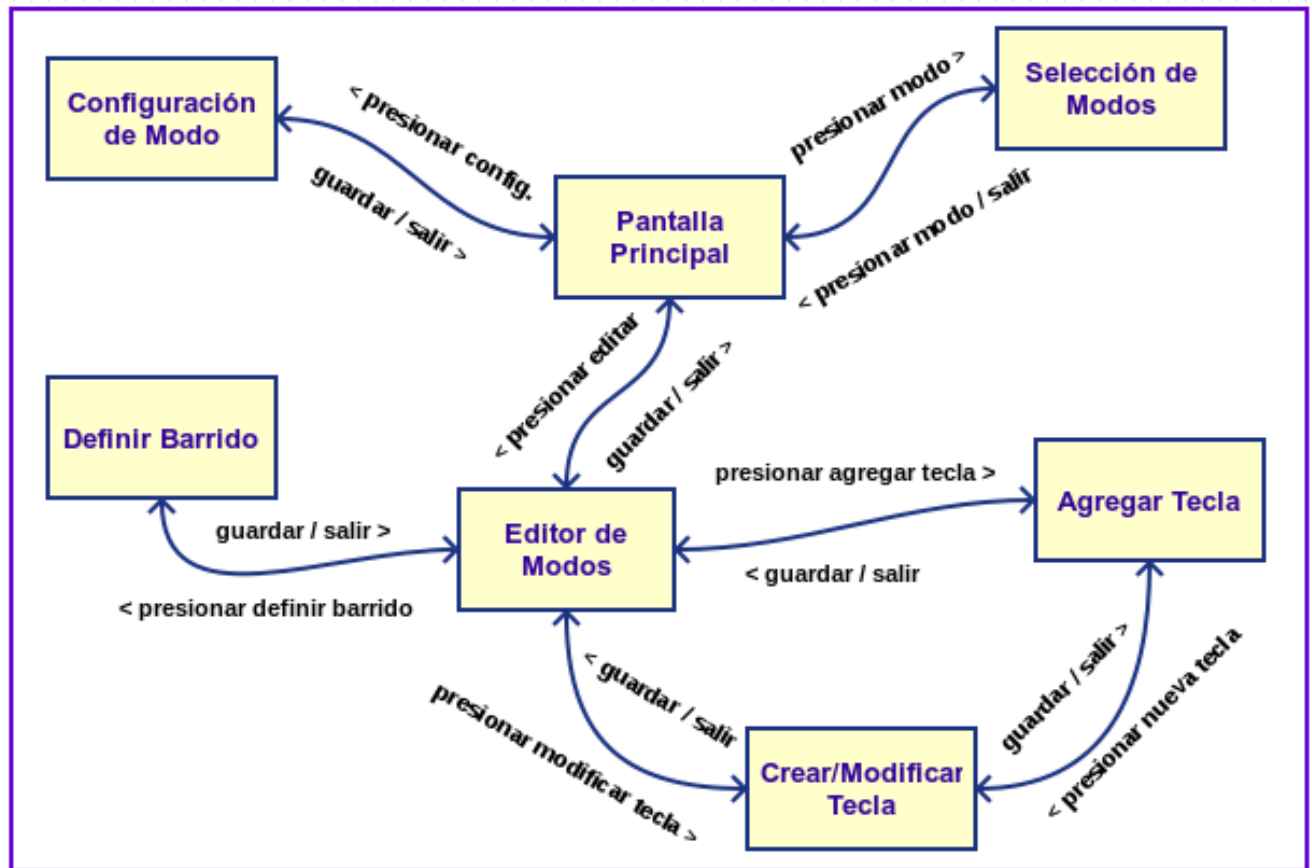


Figura 42: Navegabilidad entre pantallas

4.1.3 Actividad 3: Prototipado y Evaluación con los usuarios del diseño de GUI

Una vez que se contó con un diseño detallado se armó un prototipo utilizando la técnica de prototipado en papel y se realizó una segunda visita a la escuela para realizar una evaluación .

Segunda visita a la Escuela

En esta segunda visita como ya se dijo se trabajó sobre evaluó un prototipo en papel del diseño que realizado, la evaluación se realizó con una técnica similar al recorrido de usabilidad plural con las maestras como evaluadoras y sin un debate final entre ellas.

La evaluación se realizó con 3 maestras, se les presentó el prototipo en papel y se les explicaron las navegaciones, para que lo evaluaran dando su opinión de experto sobre qué le agregarían, que le

sacarían y que le modificarían al prototipo presentado.

En general valoraron satisfactoriamente el diseño presentado, agregando algunas cosas y modificando otras.

Estas sugerencias se describen a continuación:

- El texto que aparece en las teclas debe ir en mayúsculas. Esto en realidad no necesita modificar el diseño sino que el que crea las teclas debe ingresar el nombre en mayúscula.
- Poder cambiar de modo automáticamente al presionar una tecla. Esto le da mayor fluidez a la interacción. Por ejemplo, si se tiene un modo “zonas de la casa” con las teclas: Baño, Cocina, Cuarto y Comedor, y se escoge el Baño, que se pueda pasar automáticamente a un modo referente al Baño con actividades relacionadas: Bañarse, Lavarse los Dientes, Orinar, entre otras.

Esto implica una nueva funcionalidad.

4.1.4 Actividad 4: Análisis

Luego de la segunda visita se agregaron las modificaciones relevadas para el diseño y se pasó a un análisis y diseño ahora sí más a nivel de la lógica de la aplicación.

4.1.4.1 Nueva Funcionalidad: Cambio Automático de Modo

Se debe poder cambiar de modo al presionar una tecla, cargando un modo que esté relacionado con la tecla que se presiona. Para resolver de forma sencilla esta modificación se agrega la funcionalidad de la siguiente manera:

1. Agregar una opción al menú de la pantalla principal para poder activar y desactivar el modo automático.
2. Si al presionar una tecla, estando el modo automático activado, existe un modo con el mismo nombre que el nombre de la tecla se carga el modo con ese nombre. Esto implica agregar una navegación de la pantalla principal a la pantalla principal recargando las teclas con las del modo seleccionado.

4.1.4.2 Modelo de Dominio

A partir de la descripción de las funcionalidades relevadas, tomando el diseño de GUI como parte de esa descripción, se realizó un modelo de dominio.

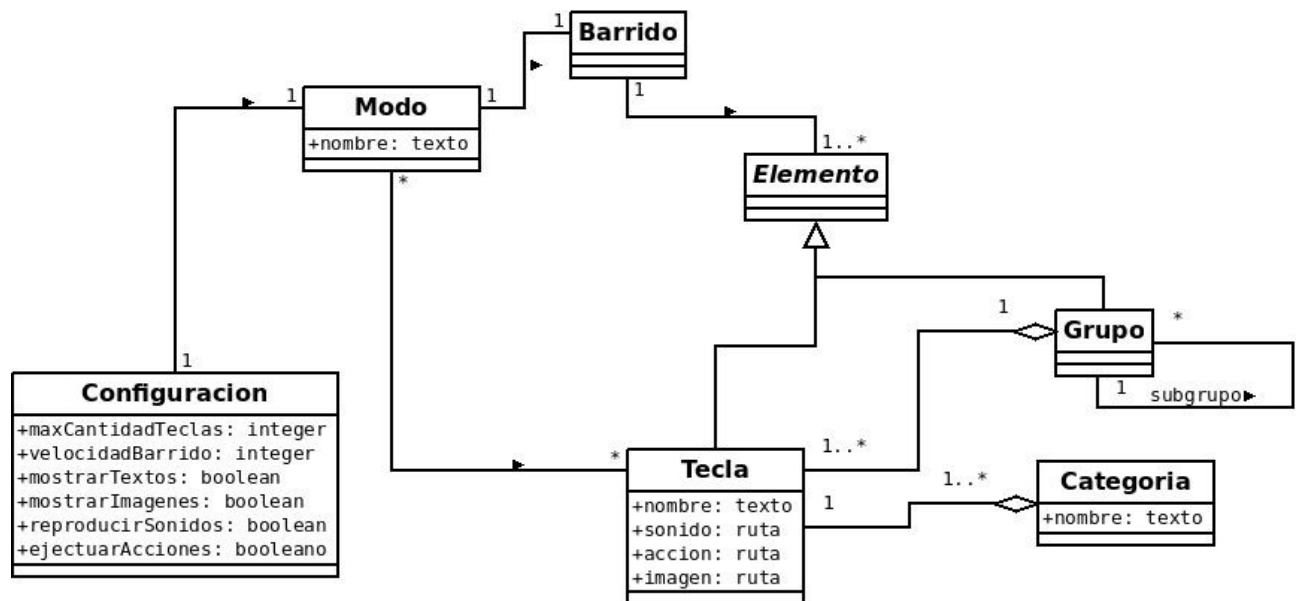


Figura 43: Modelo de Dominio

En este punto para poder continuar con el proceso de desarrollo y a su vez con la realización de las actividades del framework de evaluación de la usabilidad, se pasó a la elección de las herramientas de software que se van a utilizar para construir el prototipo basado en el diseño.

4.1.5 Actividad 5: Elección de herramientas de Software

El diseño de la lógica de la aplicación depende de qué herramientas se utilicen para construirla, si bien el lenguaje con que se va a escribir el código es Python, es necesario utilizar herramientas (plataformas, librerías, etc) que hagan más sencilla su construcción, por ejemplo para implementar la interfaz de usuario, la base de datos y la interacción con otras aplicaciones.

4.1.5.1 Implementación de Interfaz de Usuario

4.1.5.1.1 Candidatos

Para el desarrollo de la interfaz de usuario, Python tiene una gran oferta de frameworks (o toolkits). A continuación se mencionan algunos de ellos:

Tkinter - Tk

Tkinter [42] es una capa orientada a objetos que proporciona una interfaz del toolkit *Tcl/Tk* [43] para desarrollo de GUIs. *Tcl* (*Tool Command Language*) es un lenguaje de programación y scripting dinámico que cuenta entre sus utilidades del fácil prototipado y desarrollo de GUI. *Tk* es la GUI

estándar no solo de Tcl, sino también otros lenguajes dinámicos, y puede producir ricas aplicaciones que corran sin necesidad de modificaciones sobre Windows, Mac OS X, Linux y más.

Tkinter tiene la ventaja de estar incluida en la librería estándar de Python, haciéndola muy compatible a la hora de programar. Tanto *Tk* y *Tkinter* están disponibles en la mayoría de plataformas Unix, así como también Windows y Macintosh.

PyGtk - PyGObject

PyGTK [44] provee una interfaz en Python para el toolkit *GTK+* [45]. Como la propia librería *GTK+*, se encuentra actualmente bajo la licencia GNU GPL [51].

GTK+ por *GIMP Toolkit* es un toolkit multiplataforma para crear GUIs. Ofrece un set completo de elementos de interfaz conocidos como widgets en la jerga del desarrollo de GUIs. *GTK+* se ajusta al desarrollo de aplicaciones que van desde pequeñas herramientas a complejas suites.

Actualmente *PyGtk* no está siendo usada para nuevos proyectos recomendando la utilización de *PyGObject* [46] en su lugar, también las aplicaciones ya desarrolladas están siendo migradas de una a la otra. Esto se debe a que *PyGObject* funciona para la última versión de *GTK+*, *GTK+ 3* mientras que *PyGtk* soporta hasta *GTK+ 2*.

PyQt

Es la interfaz en Python para el framework multiplataforma *Qt* [47], la cual permite desarrollar de forma sencilla aplicaciones con GUIs las cuales cuentan con un aspecto nativo a la plataforma donde se ejecutan. A nivel de licencia Qt tiene licencias múltiples, si la aplicación es completamente open source se puede utilizar la versión comunitaria, en caso contrario se debe utilizar una licencia comercial.

Al igual que *PyGtk* y *PyGObject* maneja el concepto de widgets, permitiendo desarrollar aplicaciones que cuenten con los elementos clásicos de GUI de forma sencilla.

Kivy

Es una librería open source en Python que permite el desarrollo sencillo de aplicaciones que hace uso de innovadoras GUIs, como por ejemplo interfaces multitáctiles. [49]

Es multiplataforma pudiendo utilizarse en la construcción no solo de aplicaciones para Linux, Windows o Mac OS X, sino que también para sistemas operativos de dispositivos móviles como Android e iOS.

Puede utilizar de forma nativa la gran mayoría de protocolos y dispositivos de entrada tanto de computadoras como de dispositivos móviles.

También maneja el concepto de widgets y permite la creación de interfaces atractivas de forma sencilla. Además cuenta con una modelo MVC (Modelo Vista Controlador) que permite desacoplar lo que es la GUI de la lógica de negocio y la base de datos. Utilizando un lenguaje muy básico llamado *Kv Language* se pueden construir fácilmente interfaces muy atractivas que se pueden modificar sin afectar la lógica. Está bajo la licencia del MIT [50] y cuenta con un motor gráfico

construido sobre *OpenGL ES2* [52] .

4.1.5.1.2 Selección de Toolkit

Para la selección se tomaron en cuenta las siguientes características:

- **Facilidad de desarrollo:**
 - Documentación y ejemplos existentes.
 - Widgets que incluye.
- **Portabilidad:**
 - Sistemas en los que se podría utilizar la aplicación sin necesidad de cambiarla.
- **Mantenimiento:**
 - Qué tan actualizado se mantiene el toolkit y si hay una comunidad detrás que lo mantenga.
- **Licencia:**
 - Qué restricciones tiene a nivel de licencia.
- **Otras:**
 - Arquitectura del toolkit, capacidad de interacción y atractivo de las aplicaciones desarrolladas sobre él.

Por razones de licencia se descartó la opción de *PyQt*, que si bien tiene una versión comunitaria puede generar conflictos en algún futuro en caso de seguir el desarrollo de la aplicación.

PyGtk se descartó por estar siendo suplantada por *PyGObject* el cual cuenta con poca documentación. Aparte, al observar como se obligó a migrar las aplicaciones ya desarrolladas para poder utilizar *GTK +3* y no mantener la compatibilidad con versiones anteriores, puede dar la pauta que esto puede volver a suceder.

Para decidir entre *Tkinter* y *Kivy* se comparó la facilidad de desarrollo implementando un Hello World para cada uno. Ambos desarrollos fueron sencillos pero *Kivy* cuenta con una sintaxis más sencilla además de contar con el Kv Language y la arquitectura MVC. Además *Kivy* es el único que puede ser portado a dispositivos móviles y admite interacciones multitáctiles. A nivel de documentación *Kivy* cuenta con una api muy completa, ejemplos varios y a pesar de ser bastante joven tiene una comunidad bastante amplia que va en aumento.

Teniendo en cuenta los diferentes puntos mencionados se decidió utilizar ***Kivy*** como toolkit para el desarrollo de la GUI de la aplicación.

4.1.5.2 Implementación de la Base de Datos

La base de datos va implementarse en un archivo de manera de hacerlo simple. Para Python existen varias librerías para manejar este tipo de base de datos, se optó por usar *Jsonstore* [53] por ser una de las que viene incluida dentro del módulo storage de *Kivy*.

También se consideró utilizar *Shelve* [54] que es un módulo que ya viene incluido entre los módulos de Python, pero se detectó que da un problema al utilizarse con *Kivy* al momento de armar los

empaquetados para Android.

4.1.5.3 Implementación de la interacción con otras aplicaciones

La interacción con otras aplicaciones consiste en realizar las acciones que tienen asociadas las teclas, estas pueden ser:

- enviar el texto de la tecla a la ventana en que se tiene el foco de entrada de texto, como si se tratara de un teclado tradicional
- mover el mouse, hacer click y dobleclick.
- ejecutar script que realicen otras acciones como por ejemplo abrir el navegador web o apagar la computadora.

Para lograr esta interacción la aplicación a construir debe interactuar con el sistema operativo, esta interacción depende del sistema con que se esté interactuando, en el caso de la aplicación a construir, si bien lo mejor sería que continúe con la cualidad de ser multiplataforma, el objetivo es que funcione en las computadoras del Plan Ceibal, por lo que se va interactuar con sistemas *Linux*. De todas formas en la arquitectura de la solución se debe considerar tener un módulo que centralice estas interacción de manera de poder cambiar de forma sencilla el sistema con el que interactúa.

En *Linux* el manejo de eventos de los dispositivos de entrada/salida, teclado, mouse y pantalla los más comunes, se hace a través del *X Server* del *X Window System* [55], sistema encargado de mostrar la información gráfica a los usuarios.

Para Python se cuenta con la librería *Python-Xlib* [56] que permite crear clientes que se comuniquen con el *X Server*, por ejemplo para simular eventos del teclado, en particular existen extensiones como *X-Test* [57] que se encargan de simular eventos de teclado y mouse.

Para la ejecución de scripts *Python* cuenta con el módulo *Subprocess* [58] que permite crear nuevos procesos, darles parámetros de entrada y obtener su salida y códigos de retorno.

4.1.6 Actividad 6: Arquitectura y Diseño

A partir del diseño y las herramientas de software seleccionadas se define la arquitectura de la aplicación y las clases que se van a utilizar.

4.1.6.1 Arquitectura

Al utilizar *Kivy* se puede definir un arquitectura con un modelo de capas que separe la interfaz de usuario de la lógica, también se separan los módulos de manejo de la base de datos y de interacción con el sistema operativo y la base de datos queda en un archivo aparte.

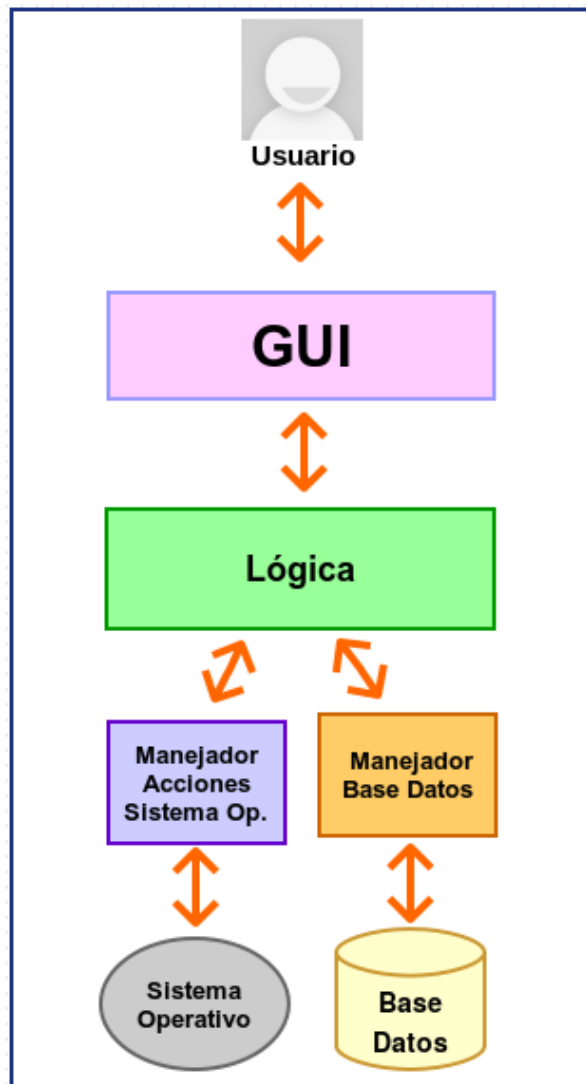


Figura 44: Arquitectura de la aplicación

4.1.6.2 Diagrama de Clases

La lógica se va a dividir en una clase por pantalla para manejar la lógica de cada pantalla por separado.

Para los manejadores de base de datos y de acciones para el sistema operativo se utilizan una clase para cada uno.

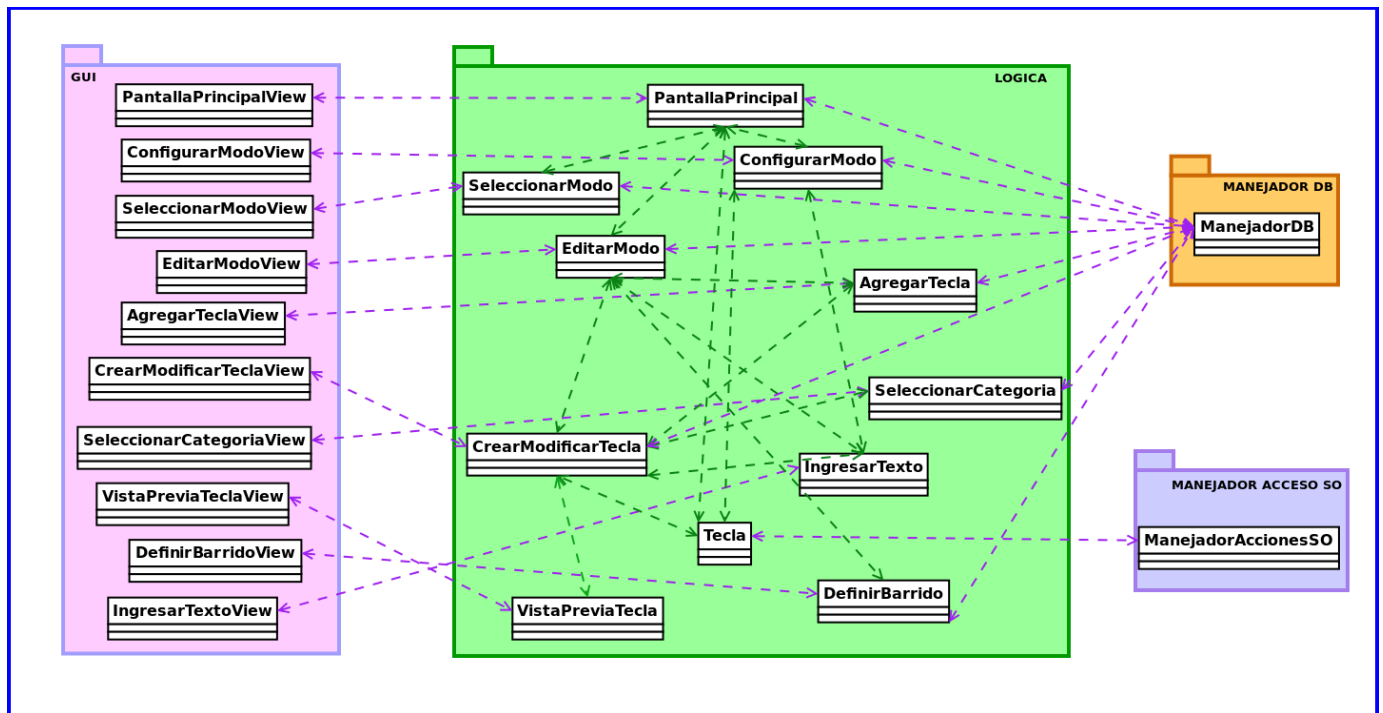


Figura 45: Diagrama de Clases

4.1.7 Fin de Iteración del Proceso de Análisis y Diseño

Una vez terminada una iteración del proceso de análisis y diseño se pasa a la implementación del prototipo con el fin de poder hacer una evaluación real de la accesibilidad y usabilidad del diseño. Una vez se obtengan los resultados de esta evaluación del prototipo se puede volver a iterar corrigiendo errores y realizando las mejoras que se consideren necesarias para volver a prototipar, evaluar, y así sucesivamente.

Capítulo 5

Implementación

En este capítulo se presentan detalles acerca de la implementación del prototipo de la aplicación.

Se incluye una descripción de algunos de los algoritmos utilizados, algunos ejemplos de implementación en Kivy y cómo se desarrolló el envío de evento al sistema operativo.

También se explica cómo se realizó el empaquetado de la aplicación para que funcione en Android.

5.1 Nombre e Icono de la Aplicación

Antes de comenzar con el contenido del capítulo se presenta el nombre y el icono de la aplicación. La aplicación se llama *comunicaME* dado que una de las principales objetivos que tiene es el de colaborar con ese desarrollo de la comunicación y la dignificación de la comprensión de los niños que se trabaja en la escuela.



Figura 46: Icono de *comunicaME*

5.2 Lenguaje y Plataforma utilizados

Como ya se mencionó en el capítulo anterior, se utilizó *Python* como lenguaje y el toolkit *Kivy* como herramienta de desarrollo de la aplicación, también se utilizaron *Jsonstore* para la base de datos, la librería *Xlib* y su extensión *XTest* y el módulo *Subprocess*.

Tanto la nomenclatura como los comentarios del código están en inglés para facilitar su divulgación y uso en un futuro.

5.2.1 Estructura del código

A continuación se presenta la estructura del código que respeta la arquitectura definida en el diseño cambiando la nomenclatura al inglés.

Archivo	Contenido
keyboard.kv	Archivo donde se define la interfaz, está dividido en las distintas views, una para cada pantalla del diseño y widgets diseñados específicamente, que se utilizan en la GUI de comunicaME
keyboard.py	Contiene la lógica de la <i>PantallaPrincipal</i> del diagrama de clases en la clase <i>KeyboardWidget</i> y es donde se crea la app de Kivy <i>KeyboardApp</i> .
mode_editor.py	Contiene la lógica del <i>EditarModo</i> del diagrama de clases en la clase <i>ModeEditor</i>
mode_scan.py	Contiene la lógica del <i>DefinirBarrido</i> del diagrama de clases en la clase <i>ModeScan</i>
mode_config.py	Contiene la lógica del <i>ConfigurarModo</i> del diagrama de clases en la clase <i>ModeConfig</i>
mode_selector.py	Contiene la lógica del <i>SeleccionarModo</i> del diagrama de clases en la clase <i>ModeSelector</i>
my_widgets.py	Contiene la lógica de varias views y widgets de uso general como la <i>InputTextView</i> para ingresar texto y <i>SelectCategoryView</i> para seleccionar una categoría, también contiene la lógica de widgets personalizados.
data_manager.py	Contiene la lógica de la conexión con la base de datos en Jsonstore.
keystore_connector.py	Contiene la lógica para la interacción con el S.O.
main.py	Contiene el main de la aplicación

5.2.2 Kivy

Una de las particularidades de *Kivy* a nivel de implementación es que cuenta con un modelo MVC y un lenguaje propio llamado *Kv Language* que permite definir la interfaz de forma independiente a la implementación de la lógica con una sintaxis muy sencilla.

A continuación se presenta un ejemplo en el que se muestra como se definió la ventana donde se selecciona el modo que se carga en la pantalla principal:

```
#####  
#####  
# MODE_SELECTOR Interface Definition  
#####  
#####  
  
<ModeSelector@ModalView>:  
    BoxLayout:  
        orientation:'vertical'  
        BoxLayout:  
            id:box_modes  
            orientation:'vertical'  
            size_hint:(1,.9)  
        Button:  
            id: btn_salir  
            text:"EXIT"  
            size_hint:(1,.1)  
            on_press:root.dismiss()
```

La sintaxis que utiliza es muy sencilla:

<ModeSelector@ModalView> : Define como una instancia de la clase **ModeSelector** va a ser representadas gráficamente. El **@ModalView** da la herencia de la clase de la clase nativa **ModalView**, clase para el widget usado para crear ventanas modales.

Luego se definen los elementos que componen el widget, estos son conocidos como sus *children*, respetando la sintaxis de Python la indentación marca la inclusión.

En el ejemplo la ventana tiene un layout que ordena sus elementos de forma vertical **orientation:'vertical'** y que tiene 2 elementos: otro layout con **id:box_modes** que se va a utilizar para contener los botones con los modos para poder seleccionar uno, y un botón con el texto **"EXIT"** que cierra la ventana modal **on_press:root.dismiss()**.

La lógica para la clase es la siguiente:

```
'''
Class that implements the logic of the mode_selector view
'''
class ModeSelector(ModalView):

    #Loads the box of modes with buttons whose names are the names of
    the existing modes
    def loadModes(self):
        dm = DataManager()
        modes = dm.getModes()
        if modes != None:
            modes_split = modes.split(',')
            box_modes = self.ids.box_modes
            for m in modes_split:
                b = Button(text=m)
                b.bind(on_press=self.selectMode)
                box_modes.add_widget(b)

    #Selects the mode whose name is the text of the button pressed
    def selectMode(self, button):
        mode = button.text
        self.attach_to.loadMode(mode)
        self.dismiss()
```

Cada widget que se defina debe contar con una clase con el mismo nombre para que ande el programa. Para poder vincular los elementos se usan las ids, en el ejemplo para poder cargar el layout de botones con los modos traídos de la base de datos, primero se obtiene el layout que tiene como id *box_modes*, `box_modes = self.ids.box_modes` y luego para cada modo se crea un botón y se agrega al layout:

```
for m in modes_split:
    b = Button(text=m)
    b.bind(on_press=self.selectMode)
    box_modes.add_widget(b)
```

A continuación un ejemplo de la pantalla de Selección de Modos, con los modos cargados: food, drinks, NUMEROS, TRANSPORTE, CLIMA, PIZZA y LLUVIA.

food
drinks
NUMEROS
TRANSPORTE
CLIMA
PIZZA
LLUVIA
EXIT

Figura 47: Pantalla de Selección de Modos

5.3 Algoritmos

5.3.1 Definición del Barrido

La forma en que se guarda el barrido de las teclas que componen un mode es la siguiente:

A cada tecla se le asocia un número que indica la posición en que se encuentra en la grilla donde se arma el teclado, las posiciones van de mayor a menor, por ejemplo, sea un modo con los números del 0 al 9:

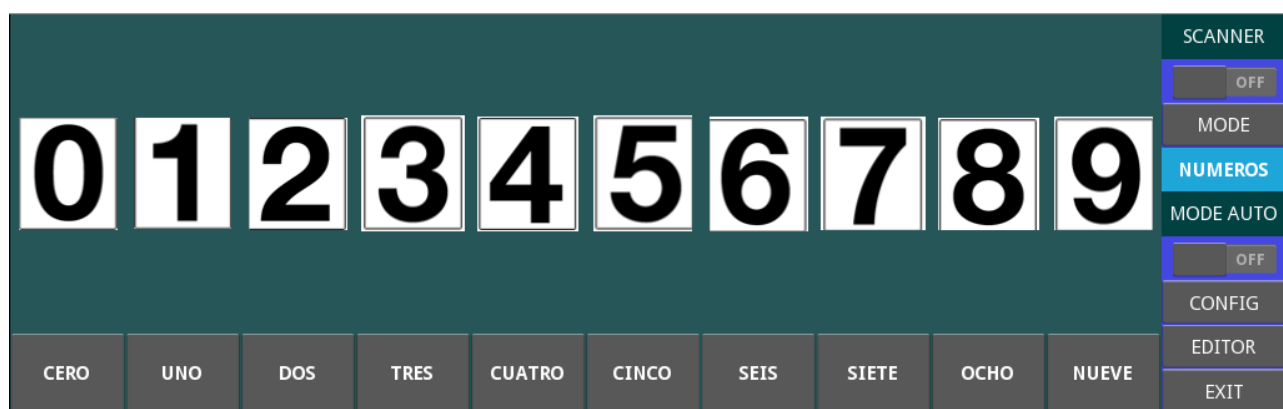


Figura 48: *comunicaME* con un modo con los números del 0 al 9

La posiciones son: cero posición 9, uno posición 8, dos posición 7, tres posición 6, cuatro posición 5, cinco posición 4, seis posición 3, siete posición 2, ocho posición 1 y nueve posición 0.

Cada grupo a nivel gráfico se define a dibujando rectángulos para cada uno de los niveles, esto se traduce en un arreglo de arreglos, por ejemplo para el modo números se define el siguiente barrido:

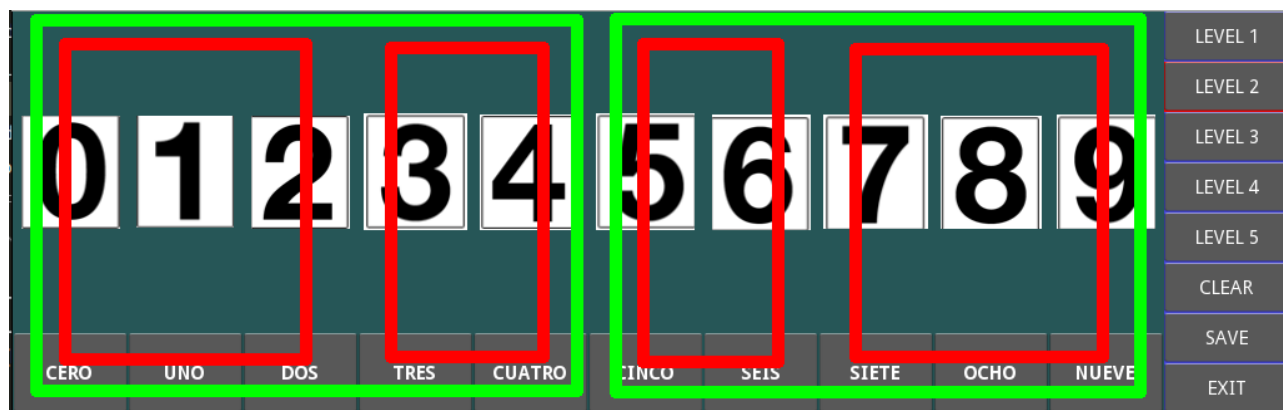


Figura 49: Barrido definido para el modo números

El algoritmo va armando este arreglo de arreglos para cada uno de los niveles, comenzando por el nivel más alto. Para simplificar la aplicación se decidió restringir la cantidad de niveles a 5. En el ejemplo los rectángulos se realizaron de izquierda a derecha y se realizaron de nivel 1 y 2, los de nivel 1 son los rectángulos en verde y los de nivel 2 son los rectángulos en rojo. Los arreglos que va armando el algoritmo son:

Como no hay de nivel 5, 4 y 3, los primeros que se arman son los de nivel 2, quedando:

- [[9], [8], [7]] primer rectángulo rojo
- [[6], [5]] segundo rectángulo rojo
- [[4], [3]] tercer rectángulo rojo
- [[2], [1], [0]] cuarto rectángulo

Recordemos que para cada tecla se pone su posición, no su texto.

Para el nivel 1 se arman los siguientes arrays:

- [[[9], [8], [7]], [[6], [5]]] el primer rectángulo verde, que incluye los 2 primeros rectángulos rojos.
- [[[4], [3]], [[2], [1], [0]]] el segundo rectángulo verde, que incluye el tercer y cuarto rectángulo.

Cuando se prende el barrido la primera zona que se barre es la del primer rectángulo verde, es decir las teclas que se incluyen en las posiciones dentro del arreglos [[[9], [8], [7]], [[6], [5]]], si se dispara un evento mientras está seleccionado se entra a barrer los de nivel 2 dentro, es decir [[9], [8], [7]] y luego [[6], [5]] .



Figura 50: Barrido del primer nivel del barrido definido.

5.3.2 Velocidad del Barrido

Para ejecutar el barrido y hacerlo variar según la velocidad configurada en la configuración del modo se utilizó la operación **schedule_once** del módulo **Clock** de **Kivy**, lo que hace es programar una tarea para ejecutarse en una cantidad de segundos indicada, en este caso la tarea es **scan2** que se encarga del barrido y el tiempo se obtiene de la configuración del modo:

```
Clock.schedule_once(self.scan2, self.getConfigModeScanSpeed())
```

Si el barrido está desactivado desde el menú de opciones se deja de programar la ejecución de la operación y se programa por primera vez al ejecutar la operación **scan**, que se ejecuta cuando se enciende el barrido desde el menú de opciones.

Este código se encuentra dentro de la clase *KeyboardWidget* en *keyboard.py*

5.4 Envío de Eventos

Para poder simular los eventos generados por el teclado y enviarlos a la ventana que tenga el foco de entrada del teclado se implementó un programa aparte que mantiene que ventana tenía el foco dado que al poner el foco en la ventana de **comunicaME** se pierde la referencia.

Esta posiblemente no sea la mejor solución dado que consume recursos extra y no se comporta de forma correcta en algunos casos, requiriendo a veces de cerrar y abrir las aplicaciones. Pero considerándose el punto más importante a probar el de la usabilidad de la interfaz de usuario, se trató de resolver la implementación de esta funcionalidad de forma sencilla y sin utilizar mucho tiempo.

Es importante en el trabajo a futuro implementar correctamente esta funcionalidad.

5.5 Empaquetando para Android

Para armar el empaquetado .apk para android se utilizó la herramienta *Buildozer* [60] que

automatiza el proceso build completo. *Buildozer* está desarrollada por los desarrolladores de Kivy.

El procedimiento una vez instalada la herramienta es muy sencillo:

Primero estando en el directorio donde se encuentra el código de la aplicación ejecutar en un terminal:

buildozer init

Esto genera el archivo **buildozer.spec** donde se cargan los parámetros de configuración de la aplicación como nombre de la aplicación, nombre del paquete, icono, versión, si se ejecuta en modo full screen, si permite la rotación de pantalla o está diseñada para mostrarse en modo landscape o portrait, son algunos de los parámetros que se pueden configurar. Una vez definidos estos parámetros ejecutar el comando:

buildozer android debug deploy

Esto genera el empaquetado que queda dentro de la carpeta genera */bin*.

5.6 Repositorio

Para el control de versiones se utilizó *Subversion* [59] , el repositorio se encuentra en:

<https://svn.riouxsvn.com/comunicame>

Capítulo 6

Test de Usabilidad

El propósito de este capítulo es describir el test de usabilidad realizado como parte del proceso de desarrollo centrado en la usabilidad utilizado, sobre el prototipo implementado *comunicaME* y validar así su diseño.

6.1 Objetivos

El objetivo general fue evaluar la usabilidad del prototipo diseñado. Esto implica descubrir en que grado el diseño de interfaz y de interacción son adecuados para el uso pretendido.

Los objetivos específicos:

- Evaluar si los niños pueden emplear esta herramienta de manera más o menos autónoma para poder comunicar ideas, deseos y necesidades, con un margen de error relativamente pequeño.
- Evaluar si las maestras pueden crear contenidos nuevos y configurar los parámetros que son modificables.

6.2 Test

6.2.1 Ubicación del Test

El testeo de usabilidad de la aplicación se realizó en una de las aulas la Escuela 200.

6.2.2 Testers

6.2.2.1 Niños

Clasificación

Debido a los distintos niveles de afección a nivel psicomotriz de los niños y niñas con P.C. se puede clasificar el espectro de usuarios según su capacidad funcional en:

- **Clase uno:** sin limitación de actividad.
- **Clase dos:** con ligera o moderada limitación de actividad.
- **Clase tres:** con limitación de la actividad, que va desde moderada hasta alta.
- **Clase cuatro:** incapacitados para desarrollar cualquier actividad física útil.

También se tiene que tomar en cuenta:

- El niño o niña tiene baja capacidad cognitiva.
- El niño o niña tiene muy comprometida la capacidad de expresarse.
- El niño o niña tiene baja visión o ceguera.
- El niño o niña tiene baja audición o sordera.

El test de usabilidad se centró en las clases dos, tres y cuatro, no incluyó casos de baja visión o ceguera ni tampoco baja audición o sordera.

Número de niños

El número total de testers fue de cuatro niños. Uno de ellos de clase dos, uno de clase tres y dos de clase cuatro . Las pruebas se efectuaron de manera individual con cada niño.

6.2.2.2 Maestras

El prototipo requiere de un usuario que cree y cargue el contenido, teniendo a las maestras de la escuela para cumplir con ese rol.

6.2.3 Materiales

- 1 tablet con el prototipo instalado y modos precargados para realizar las actividades.
- 1 computadora con el prototipo instalado y modos precargados para realizar las actividades.
- Diplomas para entregar a los testers.

6.2.4 Técnicas de Registro

- Anotaciones en Planilla de Registro.
- Videograbación.

6.2.5 Protocolo

6.2.5.1 Líneas generales

En las pruebas con ambos usuarios, se utilizó un protocolo basado en la técnica *think-out-loud* (pensar en alto). Es decir, mientras los usuarios realizan las tareas que se les proponen narran cada una de las acciones que están llevando a cabo y describen cómo están percibiendo el sistema. En el caso de los niños que no podían comunicarse, se tomaron las observaciones de sus maestras como testimonios de la experiencia.

La recolección de datos pretende registrar la eficiencia del prototipo, esto es, una vez que el usuario sabe manipularla cuánto tiempo le lleva realizar una tarea, y la eficacia, a través del tipo de errores cometidos. En este caso, el tiempo se ha tomado como una medida relativa dado que en estos casos los tiempos de reacción no suele ser una medida de validez. La satisfacción se relevó de manera informal a través de la conformidad de las maestras con la herramienta.

6.2.5.2 Actividades

Se diseñaron dos clases de actividades según las habilidades que tenían los niños. Dentro de cada actividad se proponen varias tareas. Se utilizó el mismo protocolo en cada aplicación, tanto en las instrucciones que se ofrecían de manera verbal como en la presentación de la secuencia de tareas, para evitar sesgos en el desempeño debido a la administración del test. Las tareas propuestas trataron de ser lo más representativas de las funcionalidades que tiene la herramienta ya que se

quiso explorar si alguno de los usos presentaban alguna debilidad o errores. La cantidad de tareas se estableció considerando la fatiga que podrían llegar a sentir los niños durante la sesión.

6.2.6 Descripción del protocolo

6.2.6.1 Presentación de la actividad

Se inicia la presentación de ambas actividades con una historia breve para darle un contexto divertido que ayude a que los niños se involucren y se motiven con la actividad.

Historia

Quien guía estas actividades tiene un amigo muy especial, un amigo que viene de otro planeta.

Este amigo es muy particular, no puede hablar y tiene unas manos muy grandes con un solo dedo muy grande. Al no poder hablar no se puede comunicar muy bien y le cuesta usar las computadoras, tablets o celulares que solemos usar nosotros. Sus manos resultan muy grandes y no pueden utilizarlos bien.

Por eso, quien guía estas actividades está construyendo una aplicación para que su amigo pueda comunicarse, y le gustaría que los niños lo ayuden a ver si lo que construyó está bien.

Presentación del prototipo

Luego de esta introducción, se le presenta el prototipo al niño y se explica de manera sencilla cómo funciona a través de un ejemplo. Se le muestra al niño cómo el amigo con dificultades podría pedir una pizza utilizando el prototipo. Se realizan las acciones necesarias para ello y se acompaña de una descripción verbal que apoye las acciones que se efectúan. Después de esto, se permite a los usuarios que exploren de manera libre para que realicen una inspección visual y se animen a utilizarlo con la precaución de que accedan al editor.

Actividades

Niños

Actividad 1

La actividad comienza con una explicación de los modos y del barrido. Primeramente las acciones las realiza el diseñador y a continuación se deja que el niño experimente y trate de imitar la acción realizada. La secuencia se describe a continuación:

- presentarles una imagen de comida para que vean imagen, texto y sonido.
- mostrarles varios objetos y explorarlos para dar a conocer el modo en el que se va a realizar la actividad.
- mostrarles el barrido y dejar ellos interactúen con el prototipo.

- proponerles una tarea dentro del modo. Consigna: Selecciona el medio de transporte en el viniste a la escuela. Se repite la misma consigna usando un barrido distinto.
- La actividad que se les solicita está relacionada con tareas o experiencias que hayan tenido ese día.

Actividad 2

La actividad comienza con una explicación de los modos y del barrido. Primeramente las acciones las realiza el diseñador y a continuación se deja que el niño experimente y trate de imitar la acción realizada. El objetivo con este grupo de niños es analizar el uso del teclado en modo barrido. Para ello se establecen los siguientes criterios:

- Usaremos los números para proponer un tarea. (modo: números)
- Escribimos la edad de nuestro amigo explicándole cómo funciona el barrido. La acción siempre se acompaña de una breve explicación.
- Escribimos después nuestra edad y le pedimos al niño que nos ayude de vuelta escribir nuestra edad.
- Después el niño escribe su edad usando otro tipo de barrido.

Una vez que el usuario realiza con éxito las tareas o ha pasado el tiempo suficiente, se registran las respuestas a las siguientes preguntas para ambas actividades:

1. ¿ha necesitado ayuda?
2. ¿dónde estaba la dificultad?
3. ¿esperaba otra cosa?
4. ¿que quiso hacer y no pudo?

Maestras

Las actividades que se diseñaron con las maestras lucen a continuación:

- Crear una tecla (crearlo de cero)
- Abrir un modo y agregarle el pictograma creado.
- Configurar un barrido para el modo al que se le agregó la tecla.
- Cambiar la configuración del modo:
- Sacarle las imágenes.
- Acelerar el barrido.
- Crear un modo con el nombre de la nueva tecla.

- Agregar una tecla al modo.
- Probar el cambio de modos automático.

Luego de realizadas las distintas actividades preguntar a las maestras si:

1. ¿Entendieron cómo se crean las teclas?
2. ¿Entendieron cómo crear modos?
3. ¿Entendieron cómo configurar los modos?
4. ¿Entendieron cómo definir barridos?
- 5.

6.2.6.2 Finalización de la actividad

Al finalizar las actividades se agradece a los participantes por colaborar en la creación de la aplicación y se hace entrega de un diploma.

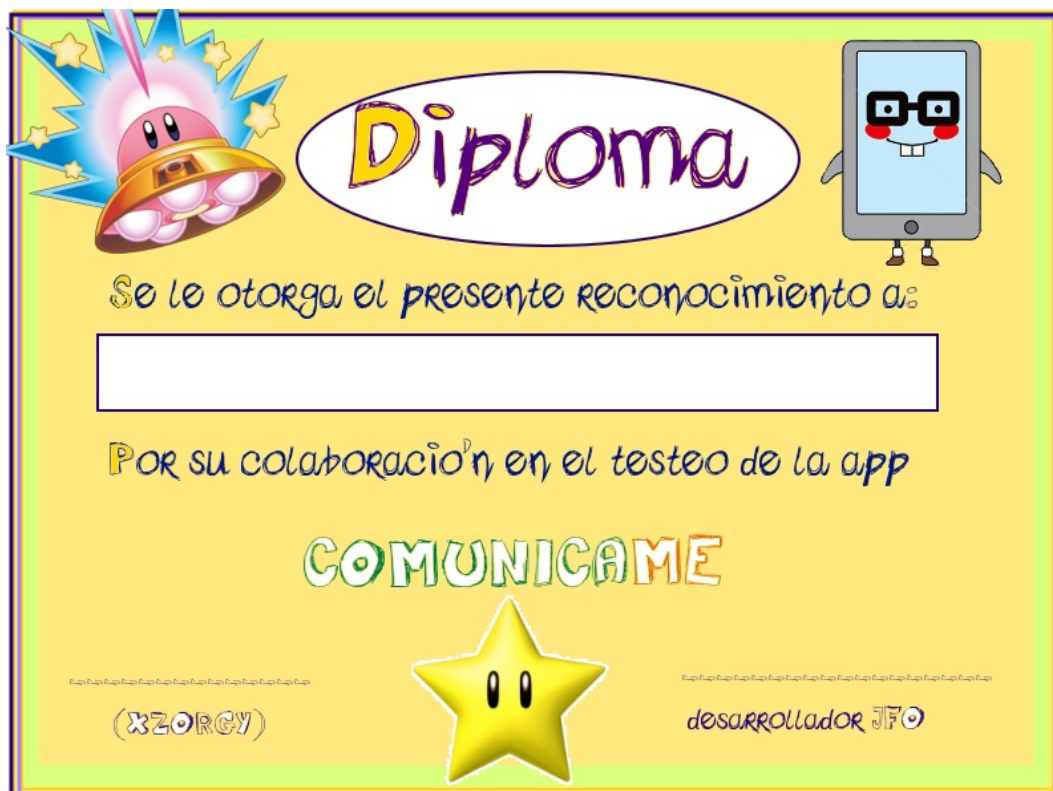


Figura 51: Diploma entregado a los niños luego de colaborar en las pruebas.

6.7 Resultados

Las pruebas se realizaron con 4 niños con diferentes patologías:

El primer tester fue Ignacio. Clasifica como Clase dos y cuenta con una buena comunicación oral. Presenta una hemiplejía espástica que le afecta el lado izquierdo, en particular la mano que tiene rigidez. Tiene 5 años de edad. No tuvo problemas para entender y realizar las dos actividades en la computadora. En uso de la tablet tuvo problemas con la pantalla multitáctil dado que presiona varias veces sin la intención de hacerlo.



Figura 52: Tester 1 realizado la actividad 2 en un laptop

La segunda fue Milagros, de clase 3. Cuenta con una comunicación oral muy limitada, tiene una diplegia espástica inferior, no puede caminar y puede utilizar las manos. No tuvo problemas en realizar las dos actividades pese que en un principio le costó entenderlas. Realizó las actividades en la computadora y en la tablet.

El tercero fue Álvaro, de Clase cuatro, y no cuenta con comunicación oral. Presenta una diplegia, no usa las manos y tiene cierto control del movimiento de los pies, aunque lo hace con dificultad es capaz de manejar la computadora. Con él se realizó la primer actividad utilizando el barrido y, una vez se logró configurar la velocidad a un nivel que le quedó adecuado, fue capaz de realizarla sin problema.

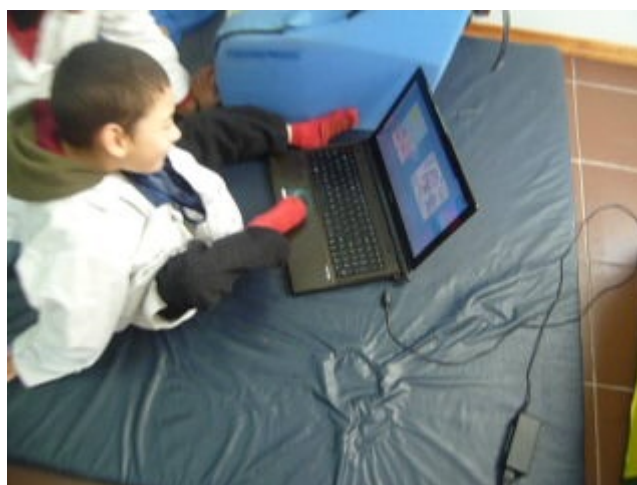


Figura 53: Tester 3 realizando actividad 1, con barrido utilizando sus pies.

El cuarto fue Facundo, de Clase cuatro. No cuenta con comunicación oral y tiene los cuatro miembros comprometidos. Utiliza la computadora a través de un pulsador. Se probaron las 2 actividades utilizando el barrido y fue capaz de realizarlas sin problema una vez que entendió cómo funcionaba el barrido.



Figura 54: Tester 4 realizando la actividad 2, utilizando barrido y un pulsador que presionaba con su cabeza.

En cuanto a las maestras se probó realizar las actividades con dos y ambas lograron entender fácilmente cómo realizar las actividades.

En suma:

El resultado del test fue positivo dado que a nivel de diseño los cuatros participantes entendieron la información que se les presentó y pudieron realizar las actividades. Los problemas detectados no afectan directamente el diseño dado que se pueden resolver sin modificarlo. Estos fue lo que se encontró:

- Problemas con la pantalla multitáctil, sobre todo cuando se utiliza la tablet con barrido, debido a que realizan varios toques que hacen que el barrido se detenga y arranque varias veces.
- Problemas con el menú de opciones, se presiona a veces sin querer ingresando a funcionalidades de edición o configuración por ejemplo sin desearlo. Para resolverlo se puede ocultar y mostrar mediante un movimiento que resulte muy difícil de realizar por error.

También a nivel de las maestras tuvo un resultado positivo las cuales mostraron un fácil entendimiento de cómo realizar las funcionalidades y le dieron una buena valoración tanto a la usabilidad como la accesibilidad del prototipo expresando su expectativa por un aplicación terminada.

En el **anexo II** se muestra la planilla utilizada para el registro de los resultados.

Capítulo 7

Conclusiones y Trabajo a Futuro

7.1 Conclusiones

Uno de los objetivos del proyecto era realizar un estado del arte que sirva para informar sobre el panorama actual de la accesibilidad y la inclusión de personas con discapacidad visto desde varios enfoques. De esta investigación se puede concluir que los gobiernos y organismos internacionales cada vez más definen y crean nuevas políticas, leyes y normativas. Esto marca la necesidad de una mayor inclusión tecnológica de las personas con discapacidad, traducéndose en la generación de nuevo hardware y software accesible o la modificación de los existentes para que también lo sean.

La oferta de herramientas tanto de hardware como de software para mejorar la accesibilidad de personas con discapacidad es amplia si se cuenta con los recursos socioeconómicos necesarios para tener acceso a ellas, dado su gran costo. A nivel de software libre si bien existen herramientas, en su mayoría son desarrolladas y mantenidas por pocos desarrolladores, por lo que son de menor calidad y se vuelven obsoletas fácilmente al dejarse de mantener.

En lo que respecta al proceso de desarrollo utilizado, seguir un proceso iterativo incremental que incluya actividades de diseño centradas en el usuario y evaluaciones continuas de accesibilidad y usabilidad del *MPIu+a*, resultó fundamental para llegar a un prototipo que cumpla con las necesidades de usabilidad de los niños de la escuela. Consideramos que seguir pautas y principios de diseño universal y diseño centrado en el usuario, evitó cometer errores de diseño que hubieran repercutido en los resultados de las evaluaciones, generando iteraciones de correcciones y retraso del desarrollo. Probablemente en este caso no se tendría un prototipo finalizado o se tendría uno de menor calidad.

En el caso particular del software orientado a niños y niñas con discapacidad motriz, el software debe poder evolucionar en su complejidad a la par que evolucionan las capacidades motoras y/o cognitivas de los niños que lo utilizan.

La experiencia de trabajo junto con los niños y maestras en el diseño de la herramienta resultó ser de gran valor. Por un lado permitió la realización de un producto más ajustado a las expectativas de los usuarios, y por el otro pensamos que influirá positivamente en la apropiación de la herramienta. Si bien este procedimiento participativo, en donde se concede una especial relevancia a las

opiniones de los usuarios, supone un coste en tiempo muy elevado (Bossen, 2002) consideramos que influirá en una reducción de costos posteriores ya que se analizaron explícitamente los modelos mentales de los niños y las maestras, esto es, cómo se imaginaban que debía funcionar o qué se podría llegar a hacer con él. Esto evitó el diseño de una solución que no fuese necesaria, que fuese inadecuada o que generase algún tipo de insatisfacción.

En cuanto a la implementación, el toolkit utilizado para el desarrollo *Kivy* es de gran utilidad para desarrollar aplicaciones ricas a nivel de interfaz de forma sencilla, con la capacidad de interacción multitáctil y portables para dispositivos móviles sin necesidad de ser modificadas. La simulación de eventos de teclado y mouse, y el manejo de ventanas y del foco, no es tarea sencilla y es una mejora que debe hacerse para lograr una aplicación usable en ese sentido a partir del prototipo.

El test de usabilidad realizado sobre el prototipo valida la usabilidad del diseño y encuentra como errores a corregir. Para el caso de los dispositivos móviles en los que se interacciona por medio de la pantalla táctil, poder desactivar el multi touch para algunos casos. Por otro lado, el poder ocultar el menú de opciones para evitar que se entren a funcionalidades de edición y configuración por error para algunos casos.

Respecto al trabajo en general se concluye que el diseño de aplicación realizado, validado a nivel de usabilidad y accesibilidad por el test realizado, sirve como base para el desarrollo completo de una aplicación que puede ser muy útil para personas con discapacidad motriz. Considerando además comentarios de las maestras manifestando la necesidad de una aplicación con las características del diseño propuesto, genera una gran motivación para terminar el desarrollo y liberar la aplicación a la comunidad para que se use, se mejore y se exploren sus posibilidades.

Como aprendizaje queda el haber utilizado un proceso desarrollo diferente al proceso de desarrollo tradicional de la ingeniería de software, el cual enseña un método de construir software muy útil, no sólo para el desarrollo orientado a la accesibilidad sino para la gran mayoría de desarrollos. Aprender pautas y principios de diseño que pueden aplicarse también al desarrollo, de manera de crear diseños que sean accesibles al mayor número de usuarios, aumentando su alcance.

A nivel personal, el trabajo con la comunidad que conforma la Escuela N 200 fue muy agradable, gratificante y motivante. Tanto los niños como las maestras, e incluso el personal, enseñan muchas cosas a nivel humano de enorme valor que van más allá de cualquier aprendizaje técnico obtenido durante el desarrollo del proyecto. La escuela siempre recibe con las puertas a quien quiera trabajar con ellos y todo el personal colabora en lo que esté a su alcance.

7.2 Trabajo a Futuro

Para lograr una aplicación a partir del prototipo lo primero es mejorar la forma en que se envían eventos de mouse y teclas, y se realizan acciones con el sistema operativo en que se está ejecutando la aplicación. El prototipo se centró en implementar y dar la posibilidad de testear el diseño realizado con los usuarios. También deben corregirse los errores detectados durante el test de usabilidad.

Durante el desarrollo fueron surgiendo también ideas de funcionalidades a agregar a la aplicación que escapaban al alcance del proyecto, a continuación se describen algunas de ellas:

(1) Interacción entre dispositivos móviles y computadoras: Poder ejecutar la aplicación en un dispositivo móvil y que interactúe con una computadora que esté en la misma red. La interacción es la misma que se plantea para el prototipo, envío de eventos de teclado y mouse, y ejecución de scripts. De esta manera el dispositivo móvil podría utilizarse como un comunicador, como un teclado y mouse en pantalla inalámbricos y como una especie de “control remoto” de la computadora con que interactúa.

(2) Poder exportar y compartir los modos definidos y las teclas creadas: Esta funcionalidad favorece la generación colectiva de una biblioteca de modos y teclas de manera de poder aprovechar mejor lo que se crea y compartirlo con la comunidad de usuarios. Esto puede llegar a generar modos de uso común en la comunidad de usuarios que aporten un lenguaje común para la aplicación. También pueden desarrollarse actividades educativas basadas en los modos creados, configuraciones de modo recomendados para determinados perfiles y barridos específicos que optimicen la utilización de cada modo.

En lo que respecta al trabajo para la accesibilidad e inclusión de las personas con discapacidad, se podría incentivar el trabajo en este tema fomentando la creación de proyectos que atiendan las necesidades de estas poblaciones. Para que esta idea se concrete, es necesario poder integrarse en las comunidades. En particular, mantener el vínculo con la Escuela 200 ofrece un escenario ideal de colaboración donde la comunidad puede expresar su necesidad y el colectivo académico puede hacerse eco, trabajando en conjunto hacia un objetivo en común. Participar en foros y mantenerse actualizado respecto a las herramientas, sobre todo a las de hardware y software libre, son medidas que pueden incrementar la calidad de las soluciones. Por último, habría que adoptar como estrategia general la adopción de los principios de diseño universal y accesible como criterios general de diseño en educación, en urbanismo, en objetos de la vida diaria, por citar algunos, sensibilizando acerca de su importancia y evidenciando el amplio rango de usuarios al que podemos llegar.

8 Referencias

- [1] “Las TIC En La Educación | Organización De Las Naciones Unidas Para La Educación, La Ciencia y La Cultura.” *Las TIC en la educación | Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura*.
<<http://www.unesco.org/new/es/unesco/themes/icts/>>
- [2] “Brecha Digital.” - *Wikipedia, la enciclopedia libre*.
<https://es.wikipedia.org/wiki/brecha_digital>
- [3] “Plan Ceibal” - *Web*
<<http://www.ceibal.edu.uy/>>
- [4] “One Laptop Per Child.” *One Laptop per Child*.
<<http://one.laptop.org/>>
- [5] “Ceibalómetro 2012.” */artículo*.
<<http://www.ceibal.edu.uy/articulo/noticias/estudiantes/ceibalometro2012>>
- [6] “Censos 2011 - Instituto Nacional De Estadística.” *Censos 2011 - Instituto Nacional de Estadística*. <<http://www.ine.gub.uy/censos2011/index.html>>
- [7] Kowtko, Marc. “Using Assistive Technologies to Improve Lives of Older Adults and People with Disabilities.” *2012 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* (2012): n. pag. 4 - 6
- [8] AbleNet – Solutions For Individuals with Disabilities.” *Default Store View*.
<<https://www.ablenetinc.com/>>
- [9] “Personas Con Discapacidad | Educación | Organización De Las Naciones Unidas Para La Educación, La Ciencia y La Cultura.” *Personas con Discapacidad | Educación | Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura*.
<<http://www.unesco.org/new/es/education/themes/strengthening-education-systems/inclusive-education/people-with-disabilities/>>
- [10] Reddihough, Dinah S, and Kevin J Collins. “The Epidemiology and Causes of Cerebral Palsy.” *Australian Journal of Physiotherapy* 49.1 (2003): n. pag. 7–12
- [11] Malagon, Jorge. “Parálisis Cerebral.” *MEDICINA (Buenos Aires)* 67.6/1 (2007): 586–592. Print.
- [12] Robaina-Castellanos, G R, S Riesgo-Rodriguez, and M S Robaina-Castellanos. “Definition and Classification of Cerebral Palsy: A Problem That Has Already Been Solved?” *Revista de neurologia* 45.2 (2007): 110–117. Print.
- [13] Argüelles, Pilar Póo. “Parálisis Cerebral Infantil.” *Asociacion Española de Pediatría*. Tabla I (2008): 271–277. Print.

- [14] Preiser, Wolfgang F. E., and Elaine Ostroff. *Universal Design Handbook*. New York: McGraw-Hill, 2001. Print.
- [15] *SIGCHI - Special Interest Group in Computer Human Interaction* - <<http://www.sigchi.org/>>
- [16] *ACM - Association for Computer Machinery* - <<https://www.acm.org/>>
- [17] *ACM SIGCHI Curricula for Human-Computer Interaction*. New York: Association for Computing Machinery, 1992. Print.
- [18] Lores J., Granollers T., Lana S., Introduction to Human Computer Interaction (Spanish), Spain: AIPO (CHI Spanish Association), 2002. Versión electrónica
<<http://aipo.es/libro/pdf/01Introd.pdf>>
- [19] Lores J., Gimeno J. , Métaforas, estilos y paradigmas, AIPO, 2001. Versión electrónica
<<http://aipo.es/libro/pdf/03Metafo.pdf>>
- [20] Weiser, Mark. "The Computer For the 21st Century." *Readings in Human-Computer Interaction* (1995): 933-940.
- [21] Azuma R., A survey of augmented reality, *Presence*, vol. 6, no. 4, pp. 355-385, 1997.
- [22] Saltiveri, Toni Granollers i, and Vidal Jesús Lorés. *MPIu+a. Una metodología Que Integra La ingeniería Del Software, La interacción Persona-Ordenador y La Accesibilidad En El Contexto De Equipos De Desarrollo Multidisciplinares*. Lleida: l'autor, 2004. Print
- [23].Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. *Proc. ACM CHI'94 Conf.* (Boston, MA, April 24-28), 152-158.
- [24] Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.
- [25] "MantisBT Makes Collaboration with Team Members & Clients Easy, Fast, and Professional." *Mantis Bug Tracker*. <<https://www.mantisbt.org/>>
- [26] "Bugzilla." *Home*. <<https://www.bugzilla.org/>>
- [27] Izzo, Margaretha Vreeburg. "Universal Design for Learning: Enhancing Achievement of Students with Disabilities." *Procedia Computer Science* 14.Dsai (2012): 343-350. Web. 22 May 2013.
- [28] "SwitchXS." *AssistiveWare*. <<http://www.assistiveware.com/product/switchxs>>
- [29] "Tobii C12 - AAC Device for Independence." *Portable speech generating AAC device (SGD)*. <<http://www.tobii.com/en/assistive-technology/global/products/old-or-discontinued-products/tobii-c12/>>

- [30] “Technology That Works in Harmony with Natural Human Behavior.” *Tobii*. <<http://www.tobii.com/>>
- [31] “Tobii CEye Eye Control Module.” *Eye controlled communication for personalized interaction*. <<http://www.tobii.com/en/assistive-technology/global/products/old-or-discontinued-products/ceye-eye-control-module/>>
- [32] “IFreeWeb.” *iFreeWeb*. <<http://www.ifreetablet.es/>>
- [33] “Grupo De Investigación EATCO.” *Grupo de Investigación EATCO*. <<http://www.eatco.es/>>
- [34] “Accessibility.” *OLPC RSS*. <<http://wiki.laptop.org/go/accessibility>>
- [35] “What Is Sugar?” - *Sugar Labs*. <http://wiki.sugarlabs.org/go/What_is_Sugar%3F>
- [36] “GNOME.” *GNOME*. <<https://www.gnome.org/>>
- [37] “How Accessibility Works In GNOME.” *How Accessibility Works in GNOME*. <<https://developer.gnome.org/accessibility-devel-guide/stable/gad-how-it-works.html.en>>
- [38] “PyAIML 0.8.5 : Python Package Index.” *PyAIML 0.8.5 : Python Package Index*. Web. 21 Aug. 2015. <<https://pypi.python.org/pypi/pyaiml>>
- [39] “Wikipedia” *Chatterbot* - <<https://en.wikipedia.org/wiki/Chatterbot>>
- [40] “Diseño De Interfaces Perceptuales Para XO.” *Diseño de interfaces perceptuales para XO*. <<http://pui-ceibalxo.blogspot.com/>>
- [41] “Pencil Project.” *Home* -. <<http://pencil.evolus.vn/>>
- [42] “Page.” *TkInter*. <<https://wiki.python.org/moin/tkinter>>
- [43] “Tcl Developer Site.” *Tcl Developer Site*. <<https://www.tcl.tk/>>
- [44] “PyGTK: GTK For Python.” *PyGTK*. <<http://www.pygtk.org/>>
- [45] “The GTK Project.” *The GTK Project*. <<http://www.gtk.org/>>
- [46] “PyGObject (Aka PyGI).” *Projects/PyGObject*. <<https://wiki.gnome.org/action/show/Projects/PyGObject?action=show&redirect=PyGObject>>
- [47] “Qt - Developer Resources - Documentation, Guides, Forums.” *Qt*. <<http://www.qt.io/developers/>>
- [48] “Page.” *PyQt*. <<https://wiki.python.org/moin/pyqt>>
- [49] “Kivy: Cross-Platform Python Framework for NUI.” *Development*. <<http://kivy.org/#home>>
- [50] “The MIT License (MIT).” *The MIT License (MIT)*. <<http://opensource.org/licenses/mit>>

- [51] “The GNU General Public License v3.0 - GNU Project - Free Software Foundation.” *The GNU General Public License v3.0 - GNU Project - Free Software Foundation*. <<http://www.gnu.org/licenses/gpl-3.0.en.html>>
- [52] “The Khronos Group - a Non-Profit Industry Consortium to Develop, Publish and Promote Open Standard, Royalty-Free Media Authoring and Acceleration Standards for Desktop and Handheld Devices, Combined with Conformance Qualification Programs for Platform and Device Interoperability.” *OpenGL ES 2_X*. <https://www.khronos.org/opengles/2_x/>
- [53] “Jsonstore 1.3.1 : Python Package Index.” *jsonstore 1.3.1 : Python Package Index*. <<https://pypi.python.org/pypi/jsonstore>>
- [54] “11.4. Shelve: Python Object Persistence” *11.4. shelve*. <<https://docs.python.org/2/library/shelve.html>>
- [55] “X.Org.” *X.Org*. Web. 21 Aug. 2015. <<http://www.x.org/wiki/>>
- [56] “The Python X Library.” *The Python X Library*. <<http://python-xlib.sourceforge.net/>>
- [57] “Darcs - Xtest.” *darcs - xtest*. <<https://bigasterisk.com/darcs/?r=xtest>>
- [58] “17.1. Subprocess — Subprocess Management.” *17.1. subprocess*. <<https://docs.python.org/2/library/subprocess.html>>
- [59] “Apache™ Subversion®.” *Apache Subversion*. <<https://subversion.apache.org/>>
- [60] “Kivy/Buildozer.” *GitHub*. <<https://github.com/kivy/buildozer>>
- [61] Pardo, Sofía, Steve Howard, and Frank Vetere. “Child-Centered Evaluation: Broadening the Child/Designer Dyad.” *Advances in Human-Computer Interaction 2008* (2008): 1–9. Web.
- [62] Normativo, Marco. “Compendio de Legislación Sobre Discapacidad. Tomo 1” Disponible en <<http://www.larediberoamericana.com/wp-content/uploads/2012/07/Compendio-leyes-discapacidad-en-AmL.pdf>>
- [63] “Discapacidad Uruguay - Portada.” *Discapacidad Uruguay - Portada*. <<http://www.discapacidaduruguay.org/>>
- [64] “Discapacidad.Gub.Uy | Portal Del Estado Uruguayo.” *discapacidad.gub.uy | Portal del Estado Uruguayo*. <<http://discapacidad.gub.uy/>>

Anexos: Anexo I

Accesibilidad educativa de niños y niñas con discapacidad: Leyes y normativas

1 Panorama Internacional [62]

1.1 Convención sobre los Derechos de las Personas con Discapacidad (2006)

En la Convención sobre los Derechos de las Personas con Discapacidad, de las Naciones Unidas, celebrada en Nueva York en agosto de 2006, se acordó: en cuanto a la cuestión fundamental de la accesibilidad, que los países identifiquen y eliminen los obstáculos y las barreras de acceso que las personas con discapacidad puedan tener para acceder, en igualdad de oportunidades que las demás personas a su entorno físico, al transporte, las instalaciones y los servicios públicos, y las tecnologías de la información y las comunicaciones(Artículo 9).

Los gobiernos deben asegurar que las personas con discapacidad puedan acceder a la educación primaria y secundaria, la formación profesional, la enseñanza de adultos y el aprendizaje permanente sin discriminación y en igualdad de condiciones que los demás. La educación debe hacer los ajustes razonables en función de las necesidades individuales, prestar los apoyos necesarios y facilitar las medidas personalizadas y efectivas en entornos que fomenten el máximo desarrollo académico y social. Emplear los materiales, las técnicas educativas, los medios y los formatos de comunicación alternativos y aumentativos para que todo el alumnado pueda alcanzar el máximo desarrollo académico y social de conformidad con el objetivo de la plena inclusión. Los alumnos y alumnas que las necesiten deben recibir las medidas de apoyo personalizadas y efectivas, con objeto de que los alumnos ciegos, sordos o sordo ciegos reciban su educación en los lenguajes, los modos y medios de comunicación más apropiados en cada caso. Se adoptarán las medidas pertinentes para emplear a maestros cualificados en lengua de signos y el Braille. La educación de las personas con discapacidad debe promover su participación de manera efectiva en la sociedad para desarrollar plenamente el potencial humano, su personalidad, talento y creatividad (Artículo 24).

1.2 Programa del Decenio de las Américas 2006 – 2016: Por los derechos y la dignidad de las personas con discapacidad

Producto de la Convención sobre los Derechos de las Personas con Discapacidad de las Naciones Unidas del 2006, ese mismo año en Perú se presentó una declaración que presenta un programa de acción para América Latina de 10 años en pro de los derechos y la dignidad de las personas con discapacidad. Esto incluye la educación.

En la misma se declara:

- Una profunda preocupación por la persistencia de las condiciones de desventaja, inequidad y discriminación en la que se encuentra la mayoría de personas con discapacidad.
- La necesidad de adoptar medidas y estrategias hemisféricas y/o regionales urgentes, que promuevan el reconocimiento y el ejercicio de todos los derechos humanos, incluidos los civiles y políticos como los económicos, sociales y culturales, así como las libertades fundamentales de las personas con discapacidad.
- El “Decenio de las Américas: por los Derechos y la Dignidad de las personas con discapacidad” durante el período 2006-2016, con el lema: “Igualdad, Dignidad y Participación”, con los objetivos de lograr el reconocimiento y el ejercicio pleno de los derechos y la dignidad de las personas con discapacidad y su derecho a participar plenamente en la vida económica, social, cultural y política y en el desarrollo de sus sociedad, sin discriminación y en pie de igualdad con los demás.
- La necesidad que durante el Decenio señalado se emprendan programas, planes y acciones para alcanzar la inclusión y la participación plena en todos los aspectos de la sociedad de las personas con discapacidad; se ejecuten programas sociales, políticos, económicos, culturales y de desarrollo, destinados al logro de oportunidades en pie de igualdad con los demás, y se promueven medidas efectivas para la prevención de nuevas discapacidades y el acceso a los servicios y programas de rehabilitación para las personas con discapacidad.

El programa propuesto tiene los siguientes objetivos:

- **Sociedad:** Contribuir a la valoración de las personas con discapacidad y a la eliminación de las barreras que impiden el desarrollo y la inclusión.
- **Salud:** Mejorar el acceso de las personas con discapacidad a los servicios de salud en igualdad de condiciones.
- **Educación:** Garantizar el acceso de personas con discapacidad a una educación inclusiva y de calidad, así como a la formación técnica y profesional que permita su inserción productiva.
- **Empleo:** Promover la inserción laboral dependiente e independiente de las personas con discapacidad en los sectores público y privado.
- **Accesibilidad:** Promover el uso del Diseño Universal para toda nueva infraestructura, eliminando las barreras físicas y de comunicación existentes.
- **Participación Política:** Asegurar el reconocimiento y ejercicio de los derechos civiles y políticos de las personas con discapacidad en todos los asuntos que interesan a la comunidad.

En particular en lo que respecta a educación propone las siguientes acciones:

- Incorporar al sistema educativo general a los niños y jóvenes con necesidades educativas especiales.
- Dotar los centros de educación con proyección inclusiva con los recursos necesarios para atender las necesidades educativas especiales de sus estudiantes.
- Asegurar la eliminación de las barreras físicas que impiden el acceso de los estudiantes con necesidades educativas especiales asociadas a su discapacidad en todos los niveles de la

- educación, como requisito para desarrollar acciones de inclusión educativa.
- Promover la capacitación continua y especializada, presencial y a distancia, de los docentes de todos los niveles educativos que favorezcan el normal desarrollo de las políticas de inclusión.
- Desarrollar adaptaciones curriculares concretas, por tipo de discapacidad, que permitan brindar una respuesta efectiva a las necesidades educativas especiales de los estudiantes con discapacidad.
- Diseñar y ejecutar programas educativos utilizando las nuevas tecnologías de la información y la comunicación para la atención de las necesidades educativas especiales asociadas a discapacidad.
- Garantizar la incorporación de contenidos curriculares relativos a temas de discapacidad que promuevan valores de respeto a la diversidad, igualdad y no discriminación.
- Priorizar y garantizar la asignación de recursos financieros suficientes para asegurar el cumplimiento de las políticas de educación inclusivas.
- Asegurar el acceso de los estudiantes con discapacidad a la formación técnica y superior como factor central de su independencia económica y social.
- Ejecutar políticas que promuevan el desarrollo y financiamiento de la investigación de temas relacionados con la discapacidad.

1.3 Estrategia del Decenio de las Unión Europea 2010 – 2020

Al igual que la Declaración de la Américas el Programa del decenio, la Estrategia del Decenio de la Unión Europea es producto de la Convención sobre los Derechos de las Personas con Discapacidad de las Naciones Unidas del 2006. Esta Estrategia proporciona un marco de acción a escala europea y nacional para abordar las distintas situaciones de hombres, mujeres y niños con discapacidad. El objetivo general de esta Estrategia es capacitar a las personas con discapacidad para que puedan disfrutar de todos sus derechos y beneficiarse plenamente de una participación en la economía y la sociedad europeas, especialmente a través del mercado único. La Estrategia identifica medidas a escala de la UE complementarias a actuaciones nacionales y determina los mecanismos necesarios para aplicar la Convención en la Unión, sin olvidar las propias instituciones de la UE. También expone el apoyo que se necesita para la financiación, la investigación, la sensibilización, la recopilación de datos y la elaboración de estadísticas.

La Estrategia se centra en la supresión de barreras e identifica ocho ámbitos primordiales de actuación con un objetivo específico para cada uno:

- **Accesibilidad:** Garantizar la accesibilidad a los bienes y servicios, en especial los servicios públicos y los dispositivos de apoyo para las personas con discapacidad.
- **Participación:** Lograr una plena participación en la sociedad de las personas con discapacidad, permitiéndoles disfrutar de todos los beneficios de la ciudadanía de la UE, suprimiendo las trabas administrativas y las barreras actitudinales a la participación plena y por igual; proporcionando servicios de calidad de ámbito local que comprendan el acceso a una ayuda personalizada.
- **Igualdad:** Erradicar en la UE la discriminación por razón de discapacidad.
- **Empleo:** Posibilitar que muchas personas con discapacidad tengan ingresos por actividades laborales en el mercado de trabajo abierto.

- **Educación y formación:** Promover una educación y un aprendizaje permanente inclusivos para todos los alumnos con discapacidad.
- **Protección social:** Promover condiciones de vida dignas para las personas con discapacidad.
- **Sanidad:** Potenciar la igualdad de acceso a los servicios sanitarios y a las instalaciones vinculadas para las personas con discapacidad.
- **Acción exterior:** Promover los derechos de las personas con discapacidad en la acción exterior de la UE.

Se determinan medidas clave respecto a cada ámbito, estos ámbitos tienen potencial para contribuir a los objetivos generales de la Estrategia y la Convención, así como a partir de los documentos políticos en esta materia de las instituciones de la UE y del Consejo de Europa, los resultados del Plan de Acción de la UE en materia de discapacidad (2003-2010) y una consulta de los Estados miembros, las partes interesadas y el público en general.

En particular en lo que respecta a la educación y la formación propone:

Sin perjuicio de la responsabilidad de los Estados miembros respecto al contenido de los planes de estudios y la organización de los sistemas educativos, respaldar el objetivo de una educación y formación inclusivas y de calidad en el marco de la iniciativa «Juventud en movimiento».

Difundir más información sobre los niveles educativos y las oportunidades que se ofrecen a las personas con discapacidad, y aumentar la movilidad de este colectivo facilitando su participación en el Programa de aprendizaje permanente.

Respaldar un marco estratégico para la cooperación europea en educación y formación, los esfuerzos nacionales encaminados, en primer lugar, a suprimir las barreras jurídicas y organizativas que se presentan a las personas con discapacidad en los sistemas generales de educación y de aprendizaje permanente; en segundo lugar, apoyar oportunamente una educación inclusiva, un aprendizaje personalizado y una identificación temprana de necesidades especiales; y, por último, a facilitar una formación y un apoyo adecuados a los profesionales que trabajan a todos los niveles educativos e informar sobre tasas y resultados de participación.

2 Panorama Latinoamericano [62]

A nivel de la región existen distintas normativas en cada uno de los estados que la integran, a continuación se presenta una tabla donde se presentan las legislaciones vigentes para los países latinoamericanos con la fechas de creación y de sus reformas.

País	Fecha	Leyes
Argentina	16 de marzo de 1981	Ley N° 22431. Sistema de protección integral de los discapacitados.
	Última Reforma: 12 de enero de 2004	Otras normas de discapacidad que modifican la Ley N° 22431
Bolivia	15 de diciembre de 1995	Ley N° 1678. Ley de Persona

		con Discapacidad.
Brasil	24 de octubre de 1989	Ley N° 7853 Sobre el apoyo a las Personas Portadoras de Deficiencia, y sobre su integración Social.
	20 de diciembre de 1999	Reglamento de Ley N° 7853 que define la Política Nacional para la integración de las personas portadoras de deficiencia.
Colombia	11 de febrero de 1997	Ley N° 361 por la cual se establecen mecanismos de integración social de las personas con limitación.
	13 de julio de 2009	Ley N° 1316 por medio la cual se reforma la Ley N° 361 de 1997.
Costa Rica	18 de abril de 1996	Ley N° 7600 de igualdad de oportunidades para las personas con discapacidad.
	23 de marzo de 1998	Decreto 26831, Reglamento a la Ley N° 7600 de Igualdad de Oportunidades para las personas con discapacidad.
Chile	5 de enero de 1994	Ley N° 19284 Establece normas para la plena integración social de personas con discapacidad.
Ecuador	6 de abril	Codificación de la Ley vigente sobre discapacidades.
Salvador	27 de abril	Ley de equiparación de oportunidades para las personas con discapacidad
Guatemala	31 de marzo de 2008	Ley de aprobación de la política nacional en discapacidad y plan de acción.
Honduras	30 de setiembre de 2005	Ley de equidad y desarrollo integral de las personas con discapacidad.
Nicaragua	27 de setiembre de 1995	Ley de prevención,

		rehabilitación y equiparación de oportunidades para las personas con discapacidad.
Panamá	Agosto de 1999	Ley N° 42 de equiparación de oportunidades para las personas con discapacidad.
Paraguay	14 de octubre de 2004	Ley N° 2479 que establece la obligatoriedad de la incorporación de personas con discapacidad a las instituciones públicas.
Perú	18 de diciembre de 1998	Ley N° 27050 de la persona con discapacidad.
Uruguay	27 de agosto de 2009	Ley N° 18651 de Protección Integral de Personas con discapacidad
Venezuela	5 de enero de 2006	Ley para personas con discapacidad.

3 Panorama Nacional [63] [64]

En Uruguay existe la La Ley N° 18551 de Protección Integral de Personas con Discapacidad desde el año 2009, en la misma se aprueban normas internacionales y nacionales sobre los derechos de las personas con discapacidad. Dichas normas tienen el propósito de promover, proteger y asegurar el goce pleno, y en igualdad de condiciones, de todos los derechos humanos por todas las personas con discapacidad sin distinción.

Los derechos fundamentales son:

1. El respeto a su dignidad humana.
2. A disfrutar de una vida decorosa lo más normal y plena posible.
3. A la adopción de medidas que le permitan lograr la mayor autonomía.
4. A la salud, la educación, la adaptación y readaptación profesionales y su inserción laboral.
5. A la seguridad económica y social, a un nivel de vida decoroso y a la vivienda.
6. A vivir en el seno de su familia o de un hogar sustituto.
7. A ser protegido contra toda explotación, todo trato discriminatorio, abusivo o degradante.
8. Al beneficio de asistencia letrada competente cuando sea indispensable para la protección de su persona y bienes. En caso de una acción judicial, el procedimiento debe ser adecuado a sus condiciones físicas y mentales.

El sistema de protección integral que establece la ley comprende:

La atención médica, educación, rehabilitación física, psíquica, social, económica y profesional y cobertura de seguridad social.

El otorgamiento de beneficios y estímulos que permitan compensar las desventajas de la

discapacidad, dándoles la oportunidad de desempeñarse en sociedad de forma equivalente a las demás personas.

Esta ley ampara no solamente a las personas con discapacidad sino que también incluye a las personas de quien ellos dependan o estén a su cuidado, entidades con personería jurídica que promuevan la prevención, desarrollo e integración de personas con discapacidad y a instituciones privadas que les proporcionen los mismos servicios que prestan a sus afiliados en general.

Para ello el estado debe prestar asistencia y tomar medidas en las siguientes áreas:

- Atención médica, psicológica y social.
- Rehabilitación integral.
- Programas de seguridad social.
- Programas tendientes a la educación en la diversidad promoviendo su integración e inclusión.
- Formación laboral o profesional.
- Prestaciones o subsidios destinados a facilitar su actividad física, laboral e intelectual.
- Transporte público.
- Formación de personal especializado para su orientación y rehabilitación.
- Estímulos para entidades que les otorguen puestos de trabajo.
- Programas educativos de y para la comunidad a favor de las personas con discapacidad.
- Adecuación urbana, edilicia y de paseo público, en áreas cerradas o abiertas.
- Accesibilidad a la informática incorporando los avances tecnológicos existentes.

Por parte del estado los organismos que actúan en materia de discapacidad la Comisión Honoraria de Discapacidad en la órbita del Ministerio de Desarrollo Social (MIDES). Dicha Comisión si bien tiene sede en Montevideo cuenta en cada uno de los departamentos con una Comisión Departamental, además de Comisiones Regionales y Subcomisiones Locales que pueden llegar a crearse en un futuro.

El MIDES en un acuerdo con la Comisión Honoraria de la discapacidad:

1. Creará un Sistema Nacional de Rehabilitación Integral en coordinación con el Ministerio de Salud Pública.
2. Desarrollará desde el Programa Nacional de Discapacidad (PRONADIS) acciones coordinadas tendientes al mejoramiento de la calidad de vida de las personas con discapacidad.
3. Promoverá la creación de hogares con internación para personas con discapacidad para las que sea imposible su atención a través del grupo familiar.

Con lo que refiere a educación, el Ministerio de Educación y Cultura (MEC) en coordinación con la Administración Nacional de Educación Pública (ANEP) debe facilitar y suministrar a la persona con discapacidad, en forma permanente y sin límites de edad, en materia educativa, física, recreativa, cultural y social, los elementos o medios científicos, técnicos o pedagógicos necesarios para que desarrolle al máximo sus facultades intelectuales, artísticas, deportivas y sociales. Desde la educación inicial se debe promover la integración de personas con discapacidad a las aulas comunes, sobre la base del reconocimiento de la diversidad como factor educativo. Se debe garantizar el acceso a la educación en todos los niveles con los apoyos necesarios, para lo cual se debe asegurar la flexibilización curricular, de los mecanismos de evaluación y la accesibilidad física y comunicacional. Se reconoce el derecho a la educación, reeducación y formación profesional orientada hacia la inclusión laboral.

Anexo II

Registro de Test de Usabilidad comunicaME

Datos Tester

Nombre:	
Nivel:	
Características:	

Preguntas

¿Entiende las teclas?
¿Le resultan adecuados los tamaños de las teclas?
¿Le resultan adecuados los colores?
¿Le resultan adecuados los tamaños de las letras?
¿Le resultan adecuado el tamaño de las imágenes?
¿Entiende el barrido?
¿Entiende que sucede en los cambios de modo?