Choix Technique et AGL

# Choix de la technologie

## Type d’application

### Client Lourd

**Avantages :**

* Un seul langage client et serveur
* Communication avec le serveur simplifié
* Rapidité de production
* Plateformes d’exécution rétro-compatibles
* Peut évoluer en client riche pour minimiser les problèmes de déploiement et de maintenance

**Inconvénients :**

* Déploiement plus lourd
* Maintenance et Mises à jour plus lourdes
* Sécurité et sauvegarde dépendantes du poste client

### Client Léger

**Avantages :**

* Coût plus faible dans le cadre d’un parc info
* Maintenance système, application centralisée et immédiatement disponible après chaque mise à jour
* Maintenance utilisateur
* Robustesse, serveur centralisé moins sensible aux pannes qu’un poste client
* Mobilité

**Inconvénients :**

* Langages client/serveur différents
* Partie cliente nécessitant plusieurs langages
* Formation des développeurs à l’intégralité de la pile technique de l’application
* Pas de mode hors-ligne, dépendant de la qualité du réseau
* Evolutions des navigateurs
* Performances en baisse si trop nombreux utilisateurs

### Conclusion : Client lourd

Nous avons donc décidé d’opter pour un client lourd pour les raisons suivantes :

* Le parc informatique dédié à la vente des maisons modulaires consistera en seulement en quinzaine d’ordinateurs portables ce qui réduira les inconvénients de déploiement et de maintenance
* L’application Madera MMB ne nécessite aucun travail collaboratif
* L’application Madera MMB nécessite d’être utilisée en mode hors connexion
* La rapidité de production d’un client lourd offrira une sécurité dans le respect du planning de développement
* Les AGL pour clients offre offrent des modules permettant le travail collaboratif

## Langage de programmation

Pour le choix du langage, nous avons mis en place une grille décisionnelle pour nous aider à déterminer le langage le plus adéquat selon une liste de critères :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Critères | Importance | Note C# |  |  |  |
| Connaissance du langage | 5 | 2 |  |  |  |
| Sécurité des données | 2 | 3 |  |  |  |
| Pérennité | 2 | 5 |  | **Légende** | **Note** |
| Performance | 3 | 4 |  | Elevé | 5 |
| Portabilité | 3 | 4 |  | Haut | 4 |
| Faisabilité | 5 | 4 |  | Neutre | 3 |
| Déploiement | 3 | 3 |  | Bas | 2 |
| Cout | 4 | 1 |  | Faible | 1 |
| Maintenabilité | 3 | 4 |  |  |  |
| Stabilité | 3 | 4 |  |  |  |
| Popularité | 5 | 5 |  |  |  |
| Normalisation | 3 | 5 |  |  |  |
| Utilisation | 3 | 4 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Critères | Importance | Note Java |  |  |  |
| Connaissance du langage | 5 | 5 |  |  |  |
| Sécurité des données | 2 | 4 |  |  |  |
| Pérennité | 2 | 4 |  | **Légende** | **Note** |
| Performance | 3 | 3 |  | Elevé | 5 |
| Portabilité | 3 | 2 |  | Haut | 4 |
| Faisabilité | 5 | 2 |  | Neutre | 3 |
| Déploiement | 3 | 3 |  | Bas | 2 |
| Cout | 4 | 3 |  | Faible | 1 |
| Maintenabilité | 3 | 2 |  |  |  |
| Stabilité | 3 | 4 |  |  |  |
| Popularité | 5 | 4 |  |  |  |
| Normalisation | 3 | 4 |  |  |  |
| Utilisation | 3 | 3 |  |  |  |

Il en sort que le C# sera certainement le langage choisi du fait de son obtention de la note la plus élevée.

# AGL :

* Visual Studio 2015